21st International Conference in Central Europe
on
Computer Graphics, Visualization and Computer Vision

in co-operation with

EUROGRAPHICS Association

WSCG 2013

Poster Proceedings

Edited by

Vaclav Skala, University of West Bohemia, Czech Republic

21st International Conference in Central Europe
on
Computer Graphics, Visualization and Computer Vision

in co-operation with

EUROGRAPHICS Association

WSCG 2013

Poster Proceedings

Edited by

Vaclav Skala, University of West Bohemia, Czech Republic

Vaclav Skala – Union Agency

WSCG 2013 - POSTER Proceedings

Editor: Vaclav Skala
c/o University of West Bohemia, Univerzitni 8
CZ 306 14 Plzen
Czech Republic
skala@kiv.zcu.cz http://www.VaclavSkala.eu

Managing Editor: Vaclav Skala

Published and printed by:
Vaclav Skala – Union Agency
Na Mazinách 9
CZ 322 00 Plzen
Czech Republic http://www.UnionAgency.eu

Hardcopy: *ISBN 978-80-86943-76-3*

WSCG 2013

International Program Committee

Benes, Bedrich (United States)
Benger, Werner (United States)
Bengtsson, Ewert (Sweden)
Bilbao, Javier,J. (Spain)
Biri, Venceslas (France)
Bittner, Jiri (Czech Republic)
Buehler, Katja (Austria)
Coquillart, Sabine (France)
Daniel, Marc (France)
de Geus, Klaus (Brazil)
de Oliveira Neto, Manuel Menezes (Brazil)
Debelov, Victor (Russia)
Feito, Francisco (Spain)
Ferguson, Stuart (United Kingdom)
Gain, James (South Africa)
Gudukbay, Ugur (Turkey)
Guthe, Michael (Germany)
Herout, Adam (Czech Republic)
Choi, Sunghee (Korea)
Chover, Miguel (Spain)
Chrysanthou, Yiorgos (Cyprus)
Juan, M.-Carmen (Spain)
Kim, HyungSeok (Korea)
Klosowski, James (United States)
Max, Nelson (United States)
Molla, Ramon (Spain)
Muller, Heinrich (Germany)
Murtagh, Fionn (United Kingdom)
Pan, Rongjiang (China)
Paquette, Eric (Canada)
Patow, Gustavo (Spain)
Pedrini, Helio (Brazil)
Platis, Nikos (Greece)
Reshetov, Alexander (United States)
Richardson, John (United States)
Rojas-Sola, Jose Ignacio (Spain)
Santos, Luis Paulo (Portugal)
Savchenko, Vladimir (Japan)
Skala, Vaclav (Czech Republic)
Slavik, Pavel (Czech Republic)
Sochor, Jiri (Czech Republic)
Sourin, Alexei (Singapore)
Sousa, A.Augusto (Portugal)
Sramek, Milos (Austria)
Stroud, Ian (Switzerland)
Szecsi, Laszlo (Hungary)
Teschner, Matthias (Germany)
Theussl, Thomas (Saudi Arabia)
Tokuta, Alade (United States)
Vitulano, Domenico (Italy)
Wu, Shin-Ting (Brazil)
Wuensche, Burkhard,C. (New Zealand)
Wuethrich, Charles (Germany)
Zara, Jiri (Czech Republic)
Zemcik, Pavel (Czech Republic)
Zitova, Barbara (Czech Republic)

WSCG 2013

Board of Reviewers

Agathos, Alexander
Assarsson, Ulf
Ayala, Dolors
Backfrieder, Werner
Barbosa, Joao
Barthe, Loic
Battiato, Sebastiano
Benes, Bedrich
Benger, Werner
Bilbao, Javier,J.
Biri, Venceslas
Birra, Fernando
Bittner, Jiri
Bosch, Carles
Bourdin, Jean-Jacques
Brun, Anders
Bruni, Vittoria
Buehler, Katja
Bulo, Samuel Rota
Cakmak, Hueseyin
Camahort, Emilio
Casciola, Giulio
Cline, David
Coquillart, Sabine
Cosker, Darren
Daniel, Marc
Daniels, Karen
de Geus, Klaus
de Oliveira Neto, Manuel Menezes
Debelov, Victor
Drechsler, Klaus
Durikovic, Roman
Eisemann, Martin
Erbacher, Robert
Feito, Francisco
Ferguson, Stuart
Fernandes, Antonio
Fuenfzig, Christoph
Gain, James
Galo, Mauricio
Gobron, Stephane
Grau, Sergi
Gudukbay, Ugur
Guthe, Michael
Hansford, Dianne
Haro, Antonio
Hasler, Nils
Hast, Anders
Hernandez, Benjamin
Hernandez, Ruben Jesus Garcia
Herout, Adam
Herrera, Tomas Lay
Hicks, Yulia
Hildenbrand, Dietmar
Hinkenjann, Andre
Chaine, Raphaelle
Choi, Sunghee
Chover, Miguel
Chrysanthou, Yiorgos
Chuang, Yung-Yu
Iglesias, Jose,A.
Ihrke, Ivo
Iwasaki, Kei
Jato, Oliver
Jeschke, Stefan
Jones, Mark
Juan, M.-Carmen
Kämpe, Viktor
Kanai, Takashi
Kellomaki, Timo
Kim, H.
Klosowski, James
Kolcun, Alexej
Krivanek, Jaroslav
Kurillo, Gregorij
Kurt, Murat
Kyratzi, Sofia
Larboulette, Caroline
Lee, Jong Kwan Jake
Liu, Damon Shing-Min
Lopes, Adriano
Loscos, Celine
Lutteroth, Christof
Maciel, Anderson
Mandl, Thomas
Manzke, Michael
Marras, Stefano
Masia, Belen
Masood, Syed Zain
Max, Nelson
Melendez, Francho
Meng, Weiliang
Mestre, Daniel,R.
Metodiev, Nikolay Metodiev
Meyer, Alexandre
Molina Masso, Jose Pascual
Molla, Ramon
Montrucchio, Bartolomeo
Morigi, Serena
Muller, Heinrich
Munoz, Adolfo
Murtagh, Fionn
Okabe, Makoto
Oyarzun, Cristina Laura
Pan, Rongjiang
Papaioannou, Georgios
Paquette, Eric
Pasko, Galina
Patane, Giuseppe
Patow, Gustavo
Pedrini, Helio
Pereira, Joao Madeiras
Peters, Jorg

Pina, Jose Luis
Platis, Nikos
Post, Frits,H.
Puig, Anna
Rafferty, Karen
Renaud, Christophe
Reshetouski, Ilya
Reshetov, Alexander
Ribardiere, Mickael
Ribeiro, Roberto
Richardson, John
Rojas-Sola, Jose Ignacio
Rokita, Przemyslaw
Rudomin, Isaac
Sacco, Marco
Salvetti, Ovidio
Sanna, Andrea
Santos, Luis Paulo
Sapidis, Nickolas,S.
Savchenko, Vladimir
Seipel, Stefan
Sellent, Anita
Shesh, Amit
Sik-Lanyi, Cecilia
Sintorn, Erik
Skala, Vaclav
Slavik, Pavel
Sochor, Jiri
Sourin, Alexei
Sousa, A.Augusto
Sramek, Milos
Stroud, Ian
Subsol, Gerard
Sundstedt, Veronica
Szecsi, Laszlo
Teschner, Matthias
Theussl, Thomas
Tian, Feng
Tokuta, Alade
Torrens, Francisco
Trapp, Matthias
Tytkowski, Krzysztof
Umlauf, Georg
Vasa, Libor
Vergeest, Joris
Vitulano, Domenico
Vosinakis, Spyros
Walczak, Krzysztof
WAN, Liang
Wu, Shin-Ting
Wuensche, Burkhard,C.
Wuethrich, Charles
Xin, Shi-Qing
Xu, Dongrong
Yoshizawa, Shin
Yue, Yonghao
Zalik, Borut
Zara, Jiri
Zemcik, Pavel
Zhang, Xinyu
Zhao, Qiang
Zheng, Youyi
Zitova, Barbara
Zwettler, Gerald

WSCG 2013

Poster Proceedings

Contents

Automatic Video Generation Using Floor Segmentation from a Single 2D Image

Geetha Kiran A
Malnad College of Engineering
India (ZIP) 573202, Hassan, Karnataka
geethaamk@gmail.com

Murali S
Maharaja Institute of Technology
India (ZIP) 570001, Mysore, Karnataka
murali@mitmysore.in

ABSTRACT

Image based video generation paradigms have recently emerged as an interesting problem in the field of robotics. This paper focuses on the problem of automatic video generation of indoor scenes that mainly consist of orthogonal planes. The algorithm infers frontier information directly from the images using a geometric context-based segmentation scheme that uses the natural scene structure in indoor environments. The presence of floor is a major cue for obtaining the termination point for the video generation. First, we perform floor segmentation using dilation and erosion methods. Second, compute the length of the floor using distance method which is used as the termination point for video generation. Finally, video is generated by cropping the image. Our approach needs no human interventions, hence it is fully automatic. We demonstrate the technique to a variety of applications, including virtual walk through ancient time images, in forensics and in architectural sites. The algorithm is tested on nearly 100 images obtained from different buildings, all of them are fairly different in interior decoration themes from each other.

Keywords
dilation, erosion, floor segmentation, floor length, video generation

1. INTRODUCTION

Video generation from a single image is inherently a challenging problem. In Imaging devices, there is a trade-off between the images (snapshots) and video because of the limitation in storage capacity. Video clips need more storage space compared to images. This motivated to generate the video from a single 2D image rather than storing video clips. Humans analyze variety of single image cues and act accordingly, unlike robots. The work is an attempt to make robots analyze similar to humans using single 2D image.

The task of generating video from photographs is receiving increased attention to generate video of architectural sites. We are addressing here the key case where dimension of the real world object or measurement of object dimension in 2D plane is unknown. However generating video using above methods is very difficult because of perspective view. Alternatively, video could be generated using proper ground known i.e., floor segmentation. In the absence of accurate measurements, we wish to exploit geometric characteristics (windows/doors) along with the color variations. Such relationships are plentiful in man-made structures and often provide sufficient information to our work. The work is well-suited for navigation on Personal Digital assistants(PDA's) and personal computers, includes cases where buildings are destroyed and only the archieve images are available. The work here is mainly carried out using indoor images. We describe a unified framework for generating video from a single 2D image.

This paper focuses on the problem of automatic video generation of indoor scenes that mainly consist of orthogonal planes. The presence of floor is a major cue for obtaining the termination point for the video generation.

Video generation using single image finds applications, including virtual walk through ancient time images, in forensics and as waiters in restaurants. It not only helps the users to enjoy the important details of the image but also provides a vivid viewing manner.

In the next section, a review on the related works is highlighted.

2. RELATED WORK

It is observed that some methods have been developed for segmentation on a single image, few which are directly relevant to the work are highlighted here. Erick Delage *et al*. have used a graph based segmentation algorithm to generate a partition of the image and assigned a unique identifier to each partition output by the segmentation algorithm in [Con00a]. Erick Delage *et al*.[Con00c] have built a probabilistic model that incorporates a number of local image features and tries to reason about the chroma of the floor in each column of the image. Ma Ling *et al.* [Con00e] have segmented the floor region automatically

 ISBN 978-80-86943-76-3

by adopting clustering analysis and also have proposed a PCA based improved version of the algorithm to remove negative effect of shadow for segmented results. Xue – Nan Cui *et al.* [Con00f] have proposed detecting and segmenting the floor by computing plane normals from motion fields in image sequences. A geometric characteristic that objects are placed perpendicular to the ground floor can be utilized to find the floor in 2D images. Surfaces often have fairly uniform appearances in texture and color and thus image segmentation algorithms provide another set of useful features which can be used in many other applications, including video generation.

A very few Researchers have proposed different methods for video generation from a single 2D image. Shuqiang jiang *et al.* [Con00g] have proposed a method to automatically transform static images to dynamic video clips in mobile devices. Xian-sheng Hua *et al.* [Con00h] developed a system named photo2video to convert a photographic series into a video by simulating camera motions. The camera motion pattern (both the key-frame sequencing scheme and trajectory/ speed control strategy) is selected for each photograph to generate a corresponding motion photograph clip. A region based method to generate a multiview video from a conventional 2-dimensional video using color information to segment an image has been proposed by Yun-Ki-Baek *et al.* [Con00i]. Na-Eun Yang *et al*.[Con00j] have proposed method to generate depth map using local depth hypothesis and grouped regions for 2D-to-3D conversion. The various methods of converting 2D to stereoscopic 3D images involves the fundamental, underlying principle of horizontal shifting of pixels to create a new image so that there are horizontal disparities between the original image and the new version. The extent of horizontal shift depends on the distance of the feature of an object to the stereoscopic camera that the pixel represents. It also depends on the inter-lens separation to determine the new image viewpoint.

The methods proposed by the authors for floor segmentation is time consuming and have made certain assumptions specific to the application. These artifacts are not of much importance in our work, this made us to propose a simple method for floor segmentation in lesser time. Using the segmented image, length of the floor could be computed by distance method. This helps in video generation.

In this paper, a method for systematically exploring an unknown bounded indoor workspace is presented.

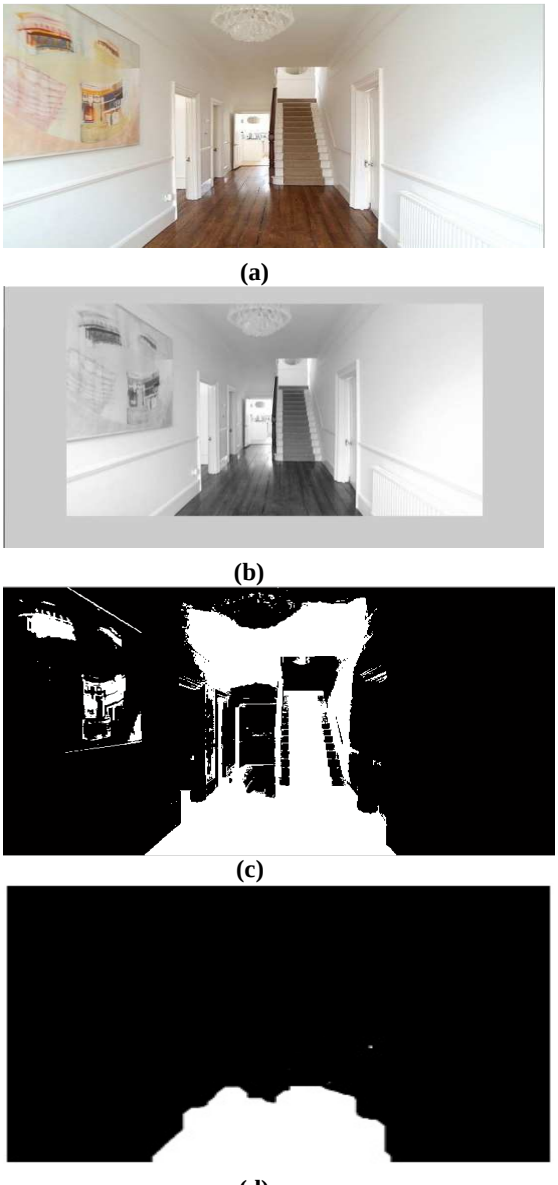
3. FLOOR SEGMENTATION

The goal is to obtain floor segmentation of a given single 2D indoor image. The crucial part of the work is detecting the pixels belonging to the floor. There are methods available for floor segmentation with known camera parameters. Requirements is to segment floor without having knowledge of camera parameters. There is possibility to find the geometric relationship, may be using color. The primary steps involves converting the given color image to gray, further convert the gray image to binary image by computing a global threshold. Finally, segment the floor by applying the dilation and erosion methods.

Segmentation:

The floor path is the major cue to generate video from a single 2D image of indoor scenes. To segment the floor from the remaining parts of the indoor image scenes, dilation and erosion techniques using the structuring elements are used.

Structuring element is used for probing and expanding the shapes contained in the input image yielding to floor segmentation. The output of the above steps is given in Figure 1.



(a) **(b)** **(c)** **(d)**

Figure 1. (a) Original Image (b) Gray Image (c) Binary image using Otsu's method (d) Segmented Image

 ISBN 978-80-86943-76-3

The segmented image obtained in Figure 1(d) is used to find the length of the floor. The distance between the start and end of the white pixel (row wise) from the floor segmented image is found by using the Euclidean distance method. This length of the floor identified could directly be used to decide the number of frames to be generated, generally 1:2 depends on the length and it can be varied with requirements. These frames are incorporated in the video generation.

4. Video Generation

The information obtained in the floor segmentation is used to generate the video. The input for the video generation are - single 2D image, computed termination point based on the distance calculated using floor segmentation, the size of the rectangle based on which cropping takes place.

The input image is considered as the first frame and the image is cropped based on the size of the predefined rectangle. Then the cropped image is resized to the original image and stored in an array of images. An appropriate set of key-frames are determined for each image based on the distance computed by using floor segmentation. The images obtained after cropping is given in Figure 2.

(a)

(b)

(c)

(d)

Figure 2. (a) Frame 1 (b) Frame 40 (c) Frame 80 (d) Frame 200

Further video is generated, using the key frames stored in the array by writing the frames to the video file. This method provides vivid dynamic effect from global view to local details.

5. EXPERIMENTAL RESULTS

The algorithm is applied to a test set of 97 images obtained from different buildings, all of them are fairly different in interior decoration themes from each other. Since the indoor images contained a diverse range of orthogonal geometries (wall posters, doors, windows, boxes, cabinets etc.), we have observed that the results presented are indicative of the algorithm performance on images of new buildings (interior) and scenes.

We also have evaluated the algorithm by manually detecting the floor path of a set of images and compared it with the floor path generated by our method. The overall accuracy obtained from the result is 91.46% as given in Figure 3.

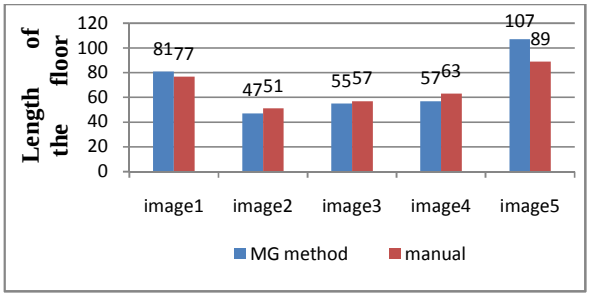


Figure 3. Comparing the length of the floor computed manually and by our method *(MG method)*

The first, intermediate and final frame generated by the method after floor segmentation is shown in Figure 4. We can observe the finer details in the intermediate and final frames that could be used in various applications including virtual walk through ancient time images, in forensics and in architectural sites.

The painting faithfully follows the geometric rules and also have color variations and therefore we can apply the methods developed here to have a virtual walk in the imaginary world.

 ISBN 978-80-86943-76-3

6. CONCLUSION

An algorithm for automatic video generation from a single 2D image is proposed and experimented for only indoor images. This paper provides a solution to transform static single 2D image into video clips. It not only helps the users to enjoy the important details of the image but also provides a vivid viewing manner. The experimental results show that the algorithm is performing well on a number of indoor scenes.

Further work can be extended to produce videos including side view, working at planar level. This requires maintenance of perspective view of the scene.

REFERENCES

[Con00a] Erick Delage Honglak Lee Andrew Y. Ng: A dynamic Bayesian network model for autonomous 3d reconstruction from a single indoor image, CVPR 2006.

[Con00b] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. IEEE Trans. Pattern Analysis and Machine Intelligence, pp.603–619, 2002.

[Con00c] Erick Delage, Honglak Lee, and Andrew Y. Ng: Automatic Single-Image 3d Reconstructions of Indoor Manhattan World Scenes, ISRR,2005

[Con00d] D. Hoiem, A. A. Efros, and M. Hebert: Geometric context from a single image. 10th IEEE International Conference on Computer Vision, 17-21 Oct, 2005.

[Con00e] Ma Ling , Wang Jianming ; Zhang Bo ; Wang Shengbei :Automatic floor segmentation for indoor robot navigation, 2nd International Conference on Signal Processing Systems (ICSPS),pp. 684 - 689, 5-7 July 2010.

[Con00f] Xue-Nan Cui, Young-Geun Kim, and Hakil Kim: Floor Segmentation by Computing Plane Normals from Image Motion Fields for Visual Navigation,International Journal of Control, Automation, and Systems, pp.788-798, 2009.

[Con00g] Shuqiang Jiang and Huiying Liu and Zhao Zhao and Qingming Huang and Wen Gao: Generating video sequence from photo image for mobile screens by content analysis, ICME, pp.1475-1478, 2007.

[Con00h] Xian-sheng Hua and Lie Lu and Hong-jiang Zhang: Automatically Converting Photographic Series into Video, 12th ACM International Conference on Multimedia,pp.708-715, 2004.

[Con00i] Yun-Ki Baek, Young-Ho Seo,Dong-Wook Kim and Ji-Sang Yoo: Multiview Video Generation from 2-Dimensional video, International Journal of Innovative Computing, Information and Control, Vol 8, Number 5(A), pp. 3135-3148, May 2012.

[Con00j] Na-Eun Yang, Ji Won Lee, Rae-Hong Park: Depth Map Generation from a Single ImageUsing Local Depth Hypothesis, 2012 IEEE ICCE,pp.311-312,13-16 Jan 2012.

[Con00k] Pedro F. Felzenszwalb and Daniel P. Huttenlocher. Efficient graph-based image segmentation. International Journal of Computer Vision, 59, 2004.

[Con00l] Young geun Kim and Hakil Kim:. Layered ground floor detection for vision based mobile robot navigation. In IEEE Robotics and Automation (ICRA), volume 1, pp 13 – 18, 2004.

[Con00m] Y. J. Jung, A. Baik, J. Kim, and D. Park, A novel 2D-to-3D conversion technique based on relative height depth cue, in Proc. Stereoscopic Displays and Applications XX, vol. 7237, Jan. 2009.

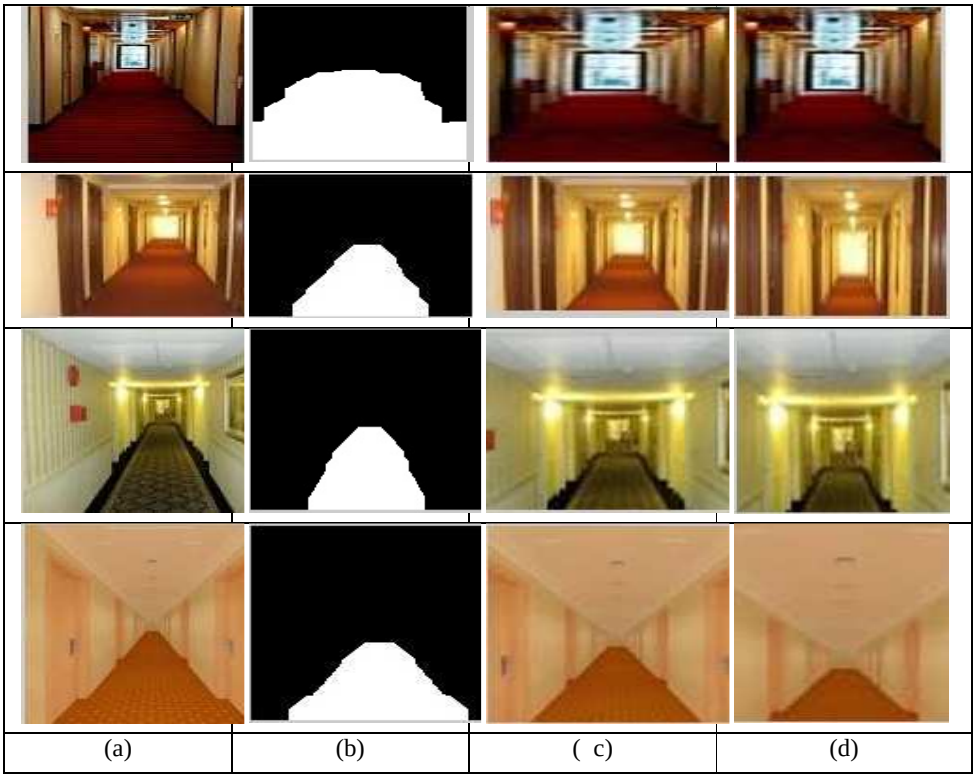


Figure 4. (a) Input Image (b) Floor Segmentation (c) Intermediate frame (d) Final Frame

CDFC: Collision Detection Based on Fuzzy Clustering for Deformable Objects on GPU's

David Mainzer
Clausthal University of Technology
Julius-Albert-Straße 4
38678 Clausthal-Zellerfeld, Germany
dm@tu-clausthal.de

Gabriel Zachmann
University of Bremen
Bibliothekstraße 1
28359 Bremen, Germany
zach@informatik.uni-bremen.de

ABSTRACT

We present a novel *Collision Detection Based on Fuzzy Clustering for Deformable Objects on GPU's* (CDFC) technique to perform collision queries between rigid and/or deformable models. Our method can handle arbitrary deformations and even discontinuous ones. With our approach, we subdivide the scene into connected but totally independent parts by fuzzy clustering, and therefore, the algorithm is especially well-suited to GPU's. Our collision detection algorithm processes all computations without the need of a bounding volume hierarchy or any other acceleration data structure. One great advantage of this is that our method can handle the broad phase as well as the narrow phase within one single framework. We can compute inter-object and intra-object collisions of rigid and deformable objects consisting of many tens of thousands of triangles in a few milliseconds on a modern computer. We have evaluated its performance by common benchmarks. In practice, our approach is faster than earlier CPU- and/or GPU-based approaches and as fast as state-of-the-art techniques but even more scalable.

Keywords

collision detection, fuzzy clustering, physics based animation, computer animation, cloth simulation

1 INTRODUCTION

Collision detection between rigid, and/or soft bodies is important for many fields of computer science . The underlying collision detection needs to check if collisions occur between a pair of objects as well as self-collisions among deformable objects. In many applications, an additional requirement is that the collision detection has to be calculated within milliseconds.

There exist various approaches that propose spatial subdivision for collision detection or approximate the surface of rigid and soft bodies. These algorithms employ axis-aligned bounding boxes (AABB), oriented bounding boxes (OBB) or Inner Sphere Trees (IST) [8] to reduce the computation time.

Most of the earlier efficient collision detection algorithms were sequential ones, which are perfect for devices that can execute only one instruction at a time. The current trend in computer architecture focuses on multi-core CPUs and many-core GPU's, and so many parallel collision detection algorithms have been proposed in the last years. The collision detection algorithm we present in this paper is a fast, fully GPU-based algorithm that can exploit data and thread-level parallelism.

1.1 Our Contributions

Our novel ***C**ollision **D**etection Based on **F**uzzy **C**lustering for Deformable Objects on GPU's* (CDFC) algorithm is designed for interactive and exact collision detection in complex environments and can handle objects movement and deformation at the same time. To achieve these features, our algorithm subdivides the whole scene into independent, overlapping parts in the first step. For the segmentation process, we use a GPU-based clustering algorithm called *fuzzy C-means*. For all clusters, we can execute the collision detection steps independently, and this offers the possibility to distribute the collision detection computation for the clusters to different GPU's.

Our novel approach is as fast as state-of-the-art collision detection algorithms but with the additional advantage that our collision detection can be distributed easily to more than only one GPU; thus it scales very well with the number of GPU's. Also, our collision detection algorithm works directly on all primitives of the whole scene, which results in a simpler implementation and can be implemented much more easily by other applications.

 ISBN 978-80-86943-76-3

2 PREVIOUS WORK

In this section, we focus on those approaches that can handle collisions between deformable objects.

2.1 Approaches Using Bounding Volume Hierarchies

Using Bounding Volume Hierarchies (BVH) is the most common approach to speed up collision detection of rigid and deformable objects. A GPU-based linear BVH (LBVH) approach was presented by Lauterbach et al. [4]. Updating these LBVH over more than one GPU is difficult and leads to a huge communication overhead.

2.2 GPU-based Collision Detection

Most modern collision detection algorithms using BVH are GPU based. However, there are some approaches which use e.g. distance fields or space subdivision to improve their performance. A hybrid CPU/GPU collision detection technique based on spatial subdivision was presented by Pabst et al. [5]. They prune away non-colliding parts of the scene by using an adapted highly parallel spatial subdivision method.

3 SWEEP-PLANE TECHNIQUE USING PCA

Due to the fact that, our collision detection approach treats all objects in a scene at the same time, we make no differences between individual objects in the rest of this paper.

During the collision detection process we use an adapted version of the standard Sweep and Prune approach, a 1D version, hereafter referred to as *Sweep-Plane* technique. We compute the bounding box for every triangle. Each bounding box spans an interval $[S_i, E_i]$ for each triangle T_i on the x-axis. Sorting all intervals along the x-axis provides information about possible colliding bounding boxes.

There exists a downside of using bounding volumes, like AABB's or OBB's. If, for example, primitives are moving then in a significant amount of cases a huge number of false positives may occur, when we choose any of the *fixed* world coordinate axes as sweep direction. In our case, the best sweep direction is the one, that allows projection to separate the primitives as much as possible. In order to achieve the best sweep direction, even if the primitives move through 3D spaces, we compute the *principal component analysis* (PCA) [2] in every frame, because the direction of the first principal component maximizes the variance of primitives, after projection.

Therefore, we move the direction of the first principal component on the x-axis. Now we compute the bounding box intervals $[S_i, E_i]$ and use the x-axis, more specifically the direction of the first component of the principal component analysis, respectively, as sweep direction. As a consequence, combining Sweep-Plane and PCA reduces the number of primitive pairs tested for intersection.

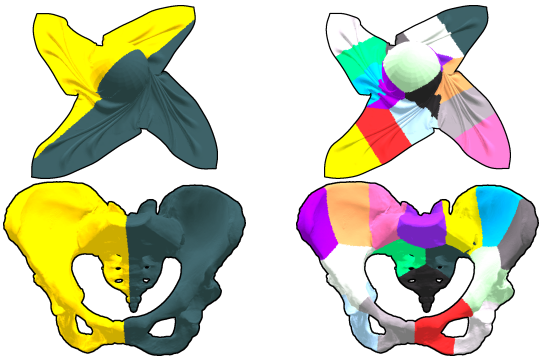


Figure 1: Examples of some high-detail objects, partitioned by fuzzy C-means into two (left column) and 16 clusters, respectively. From top to bottom: Cloth on Ball (92k) and Model of the Female Pelvis (200k).

4 OBJECT SUBDIVISION USING FUZZY C-MEANS

Using the first principal component as sweep direction only, will nevertheless produce false positives, because of the dimensional reduction in the Sweep-Plane step. The Sweep-Plane technique, used to separate the primitives, projects all 3D bounding volumes to 1D points.

To eliminate this kind of false positives we subdivide the scene (see Figure 1 for some examples) into connected components using *fuzzy C-means* (FCM) algorithm [1]. We use a fuzzy clustering algorithm because the triangles, which are located on the border between two clusters, have to be in both clusters. If adjoining clusters are not connected, then in some cases collisions across the border of the clusters would not be taken into account.

The FCM algorithm is a soft, or fuzzy, version of the well-know k-means clustering algorithm. The algorithm tries to minimize the total error, which is the sum of the squared distances of each data point to each cluster center, if we use the euclidean distance, weighted by the membership of the data point to each cluster, for all data points.

5 GPU-BASED COLLISION DETECTION

In this section we show how our method combines all previously introduced techniques. Algorithm 1 provides a short overview of the pipeline of our collision detection approach with the main procedures.

 ISBN 978-80-86943-76-3

Algorithm 1 GPU-based Collision Detection
A line represents a massively parallel computation kernel

Input: triangles of all objects
Output: intersecting triangle pairs
1: subdivide scene into c clusters using fuzzy C-means
2: **for all** clusters **do in parallel**
3: compute PCA and apply PCA
4: compute AABBs and sort AABBs along x-axis
5: collect all overlapping intervals
6: **for all** overlapping intervals **do in parallel**
7: **if** AABB intersect along y-axis **then**
8: do triangle-triangle intersection test
9: **end if**
10: **end for**
11: **end for**

First of all, we subdivide the whole scene into independent, overlapping parts by fuzzy clustering. Thus, we use the centroid of all triangles to decide to what cluster a triangle belongs to.

Now we can do the following steps for every cluster independently. As described in Section 3, we do a principal component analysis using the centroid of the triangles of the cluster. The result of the PCA is applied to the triangles of the cluster, which means that the direction of the first component of the principal component analysis points along the x-axis (step "Clustering and PCA" in Figure 3 and 4).

We are now using the x-axis as sweep plane direction because this direction maximizes the variance of primitives after projection. Therefore, we compute the bounding box of all triangles of this cluster (step "Compute AABBs" in Figure 3 and 4).

After computing the bounding boxes for all triangles of this cluster, we sort them along the x-axis using a highly-tuned Radix Sort algorithm from the Thrust[1] library.

The next challenge is to collect all bounding box intervals which intersects in the x-dimension. In order to avoid counting overlapping bounding boxes twice, we only consider the start point (S_i) of a bounding box interval. In order to receive the required memory and the position where to put all possible colliding pairs, we use the prefix sum algorithm from the Thrust library. This step, see "Collect overlapping intervals" in Figure 3 and 4, takes up the most computation time in our collision detection algorithm. The problem is that it is not possible to access the memory completely coalesced, which slows down the computation process.

After collecting all possible colliding pairs, we verify whether the bounding boxes of both triangles overlaps in the y-dimension or not. We omit an bounding box overlap test for the z-dimension, because it takes more time to read the bounding box information from memory and to compare the values, than using the triangles vertices, which may potentially needed further in the case both triangles intersect, to test if the triangles overlap in the z-dimension. If that is the case, and both triangles overlap in all three dimensions, the algorithm performs a triangle-triangle intersection test.

[1] http://thrust.github.com/

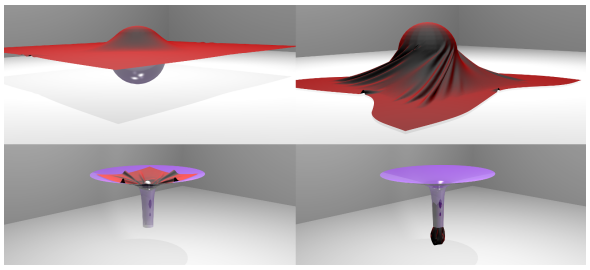


Figure 2: The upper row shows the frames 10 and 60 of the Cloth on Ball benchmark. The lower row shows the frames 125 and 375 of the Funnel benchmark.

Our collision detection algorithm compute all colliding triangle pairs and, if needed, the intersection point or line, respectively.

5.1 Accuracy & Limitations

Our collision detection algorithm will recognize every intersection between all triangles. Therefore, our approach perform bounding box intersection tests with all triangles of a cluster, to detect all colliding triangle pairs. However, in the case of significant differences in the size of the triangles, it could happen that a triangle is completely assigned to one cluster, but collides with a triangle which is completely assigned to an adjoining cluster. The reason for this is that, our approach use the centroid, which represents a triangle, for the clustering process. To prevent this, we have to decrease the membership value in the clustering step. This results in a higher degree of overlap between adjoining clusters. The size of the overlap has to be at least as large as the overall maximum distance from triangle's centroid to one of its vertices:

$$\max_{i=1,2,\ldots,n} \left(\max_{k=0,1,2} (\|C_i - vertex_{i,k}\|_2) \right) \qquad (1)$$

6 RESULTS

We have implemented our collision detection algorithm on a NVIDIA GeForce GTX 480. Therefore, we used the CUDA toolkit 5.0 as development environment. For sorting and prefix computation steps we used Thrust, a parallel algorithms library.

6.1 Benchmarking

To evaluate the performance of our collision detection algorithm in different situations, we choose some often

 ISBN 978-80-86943-76-3

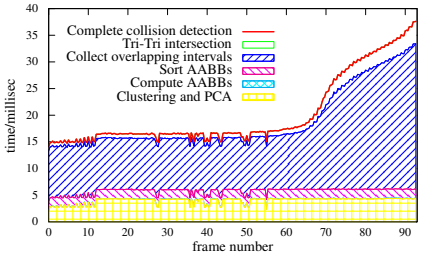


Figure 3: Collision detection time needed for Cloth on Ball (92k triangles) Benchmark.

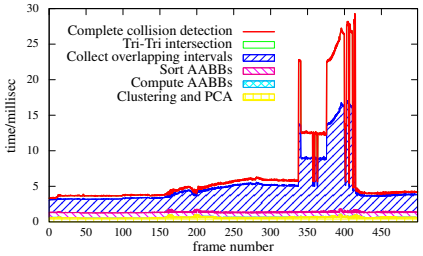


Figure 4: Collision detection time needed for Funnel (18.5k triangles) Benchmark.

used collision detection benchmarks to compare our results against other approaches.

In Table 1 we show the average collision detection time needed for all benchmarks compared with state-of-the-art collision detection algorithms.

Bench.	Our	CSt.	Pab.	HP	MC
Cl. on Ball	20.24	18.6	36.6	23.2	32.5
Funnel	6.53	4.4	6.7	–	–

Table 1: Timings (in ms) include both external and self-collision detection; CSt.[6], Pab.[5], HP[3], MC[7]

6.1.1 Cloth on Ball

In this benchmark a cloth (92k triangles) drops down on a rotating ball (760 triangles) (see Figure 2 upper row). Thereby the cloth has a huge number of self-collisions.

Figure 3 shows that the collision detection time needed to compute all collisions from frame 60 onwards increase because the number of self-collisions increase heavily like you can see on the Figure 2 (upper row). Our collision detection algorithm needs more time to collect all possible colliding triangles and has to do more intersection tests between them.

6.1.2 Funnel

A cloth (14.4k triangles) falls into a funnel (2k triangles) and passes through it, due to the force applied by a ball (1.7k triangles), who slowly increased in volume over the time (see Figure 2 lower row).

Figure 4 depicts that the collision detection time needed to compute all collisions increase slightly between frame 150 and frame 345. In these frames the cloth hit the funnel and slides a little bit into the funnel. From frame 345 onwards the ball push the cloth trough the funnel, and produces a huge number of self-collisions which results in an higher computation time needed for collision detection.

7 CONCLUSIONS

We presented a novel, accurate and fast collision detection algorithm which is completely GPU-based and needs no additional communication between host (CPU) and device (GPU) is necessary. Our *Collision Detection Based on Fuzzy Clustering for Deformable Objects on GPU's* technique can perform collision queries between rigid and/or deformable models consisting of many tens of thousands of triangles in a few milliseconds. Our results show that our collision detection algorithm is as fast as state-of-the-art approaches.

8 REFERENCES

[1] J.C. Bezdek. *Pattern recognition with fuzzy objective function algorithms*. Kluwer Academic Publishers, 1981.

[2] I. Jolliffe. *Principal component analysis*. Wiley Online Library, 2005.

[3] D. Kim, J.P. Heo, J. Huh, J. Kim, and S. Yoon. Hpccd: Hybrid parallel continuous collision detection using cpus and gpus. In *Computer Graphics Forum*, volume 28, pages 1791–1800. Wiley Online Library, 2009.

[4] C. Lauterbach, Q. Mo, and D. Manocha. gproximity: Hierarchical gpu-based operations for collision and distance queries. In *Computer Graphics Forum*, volume 29, pages 419–428. Wiley Online Library, 2010.

[5] S. Pabst, A. Koch, and W. Straßer. Fast and scalable cpu/gpu collision detection for rigid and deformable surfaces. In *Computer Graphics Forum*, volume 29, pages 1605–1612. Wiley Online Library, 2010.

[6] M. Tang, D. Manocha, J. Lin, and R. Tong. Collision-streams: fast gpu-based collision detection for deformable models. In *Symposium on Interactive 3D Graphics and Games*, pages 63–70. ACM, 2011.

[7] Min Tang, Dinesh Manocha, and Ruofeng Tong. Mccd: Multi-core collision detection between deformable models using front-based decomposition. *Graphical Models*, 72(2):7–23, 2010.

[8] Rene Weller and Gabriel Zachmann. Inner sphere trees for proximity and penetration queries. In *Robotics: Science and Systems Conference (RSS)*, Seattle, WA, USA, June/July 2009.

 ISBN 978-80-86943-76-3

Simultaneous Skeletonization and Topologic Decomposition for Digital Shape Reconstruction

Tilman Wekel and Olaf Hellwich
TU-Berlin
Marchstrasse 23
Germany, 10587 Berlin
{olaf.hellwich|t.wekel}@tu-berlin.de

ABSTRACT

Analyzing the structure of 2D shapes has been studied intensively in recent years. It is a key aspect in various computer vision and computer graphics applications. In this paper, a new algorithm is proposed which efficiently computes a skeleton and a corresponding decomposition of an arbitrary shape. Given the Voronoi diagram, the pruning step has linear complexity. The skeleton is a sparse 1D representation which captures the topology as well as the general structure of a shape. Considering the Voronoi diagram of the boundary vertices, the skeleton is extracted as a subset of the Voronoi edges using a simple classification scheme. A parameter allows to control the skeleton's sensitivity to perturbations in the boundary curve. The dual Delaunay triangulation yields a topological decomposition of the shape that is consistent with the skeleton. Each part can be classified as belonging to one of three base types which have some interesting properties. The method has been successfully implemented and evaluated. The presented concepts can also be applied to manifold surfaces which is particularly useful for digital shape reconstruction as it is shown at the end of this paper.

Keywords

skeletonization, shape decomposition, topology, Voronoi diagram

1 INTRODUCTION

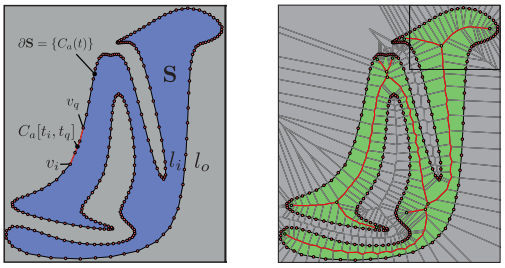
A skeleton is a 1D representation of a shape that can be intuitively compared to a stick figure. The skeleton captures the topological structure of a shape. It is used as a compact descriptor for various applications such as shape classification or geometric modeling. According to Andrés et al., related approaches can be associated to one of three categories: driven by shape thinning, distance transform-based and methods that use the Voronoi diagram of the boundary points [SM12]. Thinning algorithms iteratively shrink the shape until it is degenerated to a line. The method is commonly applied in digital image analyses, where the discrete nature of the data allows a precise definition of connectivity and thinness. Recently, the idea has been transferred to 3D voxel data [HL10]. Thinning algorithms often have a lack in performance or produce disconnected skeletons. They typically work on discrete pixel-based data and it would be difficult to adapt them to continuous domains. Aicholzer et al. introduce the straight skeleton which is constructed by contracting the boundary towards the interior of the shape [Aic96][CV12]. Distance transform-based algorithms compute a distance map for the interior region of the shape. The map associates a scalar value to each point, where the scalar value is the distance to the closest boundary point [RS04]. These methods allow a geometrically accurate construction of the medial axis. In a pruning step, unwanted branches are deleted while the topology of the remaining skeleton needs to be preserved [XB07]. Amenta et al. uses the medial axis to reconstruct a surface from a 3D point-cloud [NA01]. Voronoi-based methods evolve the fact, that the Voronoi diagram of the boundary points consists of edges that are bisectors of opposing boundary points. In the approach of Ogniewicz et al., the relevance of each edge is computed based on the maximally filling disk and the chord length of the corresponding boundary segment. Compared to our approach, the algorithm invokes computationally expensive processing steps such as the estimation of the maximally filling disk. In a similar approach, Brandt and Algazi introduce a method to estimate the minimum sampling density for the boundary points [Bra92]. Beeson et al. use the minimum spanning tree for the pruning step [PB05]. Algorithms for shape decomposition are often inspired by psychology and human perception [JL06b]. The work proposed by Lien et al. efficiently decomposes arbitrary shapes in approximately convex shapes [JL06a]. Rosin proposes a partitioning scheme based on a simple convexity measure which is calculated as the ratio of the area of the shape itself and the area of its convex hull [Ros00]. The decomposition scheme presented by Liu et al. is based on convexity and Morse theory [HL10]. Lien et al. presents an approach, where a skeleton and a decomposition are computed alternatively [JL06b]. Our approach yields some significant advantages. Given an

 ISBN 978-80-86943-76-3

arbitrary shape and its Voronoi-diagram, the pruning is computed in linear time with respect to the number of boundary points. The presented scheme elegantly connects skeleton and decomposition based on the duality between Voronoi diagram and Delaunay triangulation. Based on the work of Aigner et al., the shape is decomposed into a set of topologically meaningful parts which is useful for digital shape reconstruction, registration or classification [WA12]. Compared to Aigner et al., skeletonization and decomposition is computed simultaneously and the estimation of a maximally filling disk is not required.

2 SKELETON EXTRACTION

As stated, the skeleton is a suitable 1D-representation of a shape. Clearly, there is a relation to the Voronoi diagram, where each Voronoi edge is an equidistant bisector of two points. In the following it is shown, how the skeleton can be constructed based on the Voronoi diagram of the boundary points. Consider an arbitrary shape $\mathbf{S} \in \mathbb{R}^2$, which is given by a set of closed, continuous and non-intersecting boundary curves $\partial\mathbf{S} = \{C_0,\ldots,C_n\}$. For the sake of simplicity, a curve is assumed to be given in a parametrized form: $C_i(t_i), t_i \in [0\ldots1]$. Each curve of $\partial\mathbf{S}$ is approximated by a set of points $\mathbf{V} = \{V_0,\ldots,V_n\}$. All points in $V_a \in \mathbf{V}$ lie on the respective curve: $\{\forall v_i \in V_a : v_i = C_a(t_i)\}$, where t_i is the associated parameter. A given point $v \in \mathbb{R}^2$ is either classified as lying inside or outside the shape, $\mathscr{L}(v) \in \{l_I, l_O\}$. Consider an open curve segment which is defined to be the shortest curve between two vertices lying on the same curve: $C_a[t_i,t_q], \{v_i,v_q\} \in V_a$. The situation is visualized in (a) of Figure 1. The arc length of this curve segment can be computed by: $L(C_a[t_i,t_q]) = \int_{t_i}^{t_q} \|\dot{C}(t)\| dt$. According to Ogniewicz et al., curve potentials are an elegant way to calculate the arc length of an arbitrary curve segment [RO95]. Given a curve C_a, an arbitrary reference point v_0 is defined. The curve potential of a point v_i with the corresponding parameter t_i is given by: $\phi_a(v_i) = L(C_a[v_0,v_i])$. $\phi_a(v_i)$ needs to be calculated only once for each boundary point. A weight function $\omega(\cdot,\cdot)$ that corresponds to the arc length of a curve segment is introduced as:



(a) (b)

Figure 1: Shape boundary (a) and Voronoi diagram (b).

$$\omega(v_i,v_q) = \begin{cases} \|\phi_a(v_i) - \phi_a(v_q)\| & \text{if } v_i, v_q \in V_a \\ \infty & \text{otherwise.} \end{cases} \quad (1)$$

It can be efficiently calculated based on a difference of potentials. Note that the function $\omega(\cdot,\cdot)$ is defined to be infinity if the vertices do not belong to the same curve. The boundary vertices V define the sites of a Voronoi diagram D. The Voronoi diagram consists of vertices, edges, sites and regions: $D = \{\tilde{V},\tilde{E},\tilde{S},\tilde{R}\}$. All elements of the Voronoi diagram are indicated by a tilde. The sites are given by the boundary vertices. Let $\tilde{s}_i = v_i$ and $\tilde{s}_j = v_j$ be two sites, then all points which are closer to $\tilde{s}_i$ than to $\tilde{s}_j$ can be defined as: $\tilde{r}_{ij} = \{p \in \mathbb{R}^2 : \delta(p,\tilde{s}_i) < \delta(p,\tilde{s}_j)\}$. Where $\delta(.,.)$ is the distance function. The corresponding Voronoi region $\tilde{r}_i$ is then the set of all points which are closer to $\tilde{s}_i$ than to any other site: $\tilde{r}_i = \bigcap_{i \neq j} \tilde{r}_{ij}$. The Voronoi regions meet at Voronoi edges which can be seen as equidistant bisectors of their adjacent sites. A plot of the problem can be seen in (b) of Figure 1. The skeleton $\tilde{F}$ is defined as a subset of the Voronoi diagram. $\tilde{F}$ is a graph structure that consists of vertices and edges, $\tilde{F} = \{\tilde{V},\tilde{E}\}$. A Voronoi edge $\tilde{e}_o$ is associated with its corresponding source and target vertex $\{\tilde{v}^s(\tilde{e}_o), \tilde{v}^t(\tilde{e}_o)\}$ and its right and left site $\{\tilde{s}^r(\tilde{e}_o), \tilde{s}^l(\tilde{e}_o)\}$. $\tilde{e}_o$ is classified as belonging to the skeleton $\tilde{e}_o \in \tilde{F}$, if the following conditions are fulfilled:

1. The Voronoi edge must have a source and a target vertex: $(\tilde{v}^s(\tilde{e}_o) \neq \emptyset) \cap (\tilde{v}^t(\tilde{e}_o) \neq \emptyset)$.
2. The source and the target vertex must lie inside the shape: $(\mathscr{L}(\tilde{v}^s(\tilde{e}_o)) = l_I) \cap (\mathscr{L}(\tilde{v}^t(\tilde{e}_o)) = l_I)$.
3. Consider the boundary points that correspond to the sites: $\tilde{s}^r(\tilde{e}_o) = v_r, \tilde{s}^l(\tilde{e}_o) = v_l$. The weight of the points has to be larger than a defined threshold: $\omega(v_r,v_l) > \omega_{min}$.

The edge classification scheme is visualized in (a) of Figure 2. The first and second condition imply that the resulting skeleton lies inside the corresponding shape. The classification can be performed for each edge which implies linear complexity $\mathscr{O}(n)$ with respect to the number of Voronoi edges. The computation of the Voronoi diagram has a complexity of $\mathscr{O}(n \log n)$.

3 SEGMENT GRAPH

Given a skeleton $\tilde{F} = \{\tilde{V},\tilde{E}\}$, each vertex is associated with a degree $d(\tilde{v})$ that gives the number of adjacent skeleton edges. According to the properties of the Voronoi diagram, the degree of a vertex $\tilde{v} \in \tilde{F}$ is either one, two or three, $d(\tilde{v}) \in \{1,2,3\}$. Edges which

 ISBN 978-80-86943-76-3

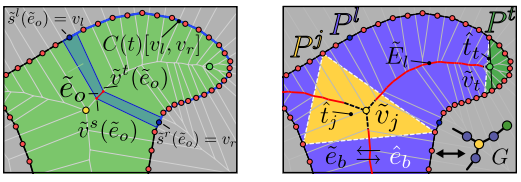
are adjacent to at least one vertex with a degree not equal to two are defined as bridges: $\tilde{E}_b$: $\{\forall \tilde{e}_o \in \tilde{E}_b : d(\tilde{v}^s(\tilde{e}_o)) \neq 2 \cup d(\tilde{v}^t(\tilde{e}_o)) \neq 2\}$. $\tilde{E}_b$ implies a decomposition of the skeleton, such that each part only contains vertices of the same degree. A is introduced as the Delaunay triangulation which is dual to the underlying Voronoi diagram of the boundary points: $D \leftrightarrows A$. All elements of the Delaunay triangulation are indicated by a hat. A bridge $\tilde{e}_b \in \tilde{F}$ is now associated with its dual Delaunay edge: $\tilde{e}_b \leftrightarrows \hat{e}_b$. $\hat{e}_b$ is referred as separator edge. $\hat{e}_b$ connects two boundary vertices and splits the shape into two parts. The set of all bridges and their corresponding separator edges decomposes the skeleton and the shape into a set of three different types of parts: $\mathbf{S} = \{P_0{}^{j,t,l}, \ldots, P_m{}^{j,t,l}\}$:

1. *Junction*. A junction P^j is given by a single vertex $\tilde{v}_j, d(\tilde{v}_j) = 3$ and its dual Delaunay triangle: $\tilde{v}_j \leftrightarrows \hat{t}_j$. $\hat{t}_j$ coincides with the three adjacent separator edges.

2. *Terminal*. A vertex with the degree of one $\tilde{v}_t, d(\tilde{v}_t) = 1$ yields a terminal part P^t. Consider the dual Delaunay triangle: $\tilde{v}_t \leftrightarrows \hat{t}_t$. The boundary of P^t is now given by the adjacent separator edge $\hat{e}_t$ and the curve segment of the corresponding boundary curve that passes through the vertices of $\hat{t}_t$.

3. *Linear*. A linear part P^l represents a branch that is defined by a linear sequence of edges, $\tilde{E}_l$: $\{\forall \tilde{e}_o \in \tilde{E}_l : d(\tilde{v}^s(\tilde{e}_o)) = d(\tilde{v}^t(\tilde{e}_o)) = 2\}$. The adjacent separator edges define a left and a right segment on the corresponding boundary curves.

Now, G is introduced as the segment graph, where the set of all parts represents the nodes of G. The edges of G are given by the bridges and their respective separator edges. The resulting decomposition into junctions (yellow), terminal (green) and linear parts (purple) as well as the corresponding segment graph is visualized in (b) of Figure 2. The dashed lines represent the bridges (black) and the separator edges (white).



(a) (b)

Figure 2: Topology of a Voronoi edge (a) and semantic decomposition (b).

4 EVALUATION AND APPLICATION

In this section, the presented algorithm is applied to various shapes which are taken from the MPEG-7 library [TZ01]. The decomposition as well as the skeleton is compared with competitive algorithms [JL06a][PB05]. The results are shown in Figure 3. The skeleton computed based on the approach of Beeson et al. can be found in (a), the convex shape decomposition corresponds to (b) and the combined outcome of our method is to be found in (c). The skeleton in (c) clearly reflects the most important features of the boundary curve while unimportant details are ignored. Each junction in the skeleton yields a triangle (yellow) which allows to decompose the shape into compact and meaningful parts along its topology. The approach based on approximate convexity has several disadvantages. The overall structure of the decomposition is very sensitive to small perturbations in the boundary curve as it can be seen in (1,b). While the shape of each spike of the star is almost similar, the decomposition varies significantly. Moreover, the decomposition is often characterized by a bad scaling (3,b). Some small regions such as the neck or the upper part of the hind legs are heavily oversegmented. The approach presented by Beeson et al. is designed for grid-based data structures only. The resulting skeleton appears to be quite sensitive to small deformations of the shape as it can be seen in (1,a). The inner part of the skeleton looks not very symmetrical while the shape itself has a symmetric structure. The skeleton in (c) looks much smoother especially at vertices where multiple branches intersect. The presented approach has been developed in the context of digital shape reconstruction, where CAD (Computer aided design) models are reconstructed from triangulated surfaces. Man-made objects can be seen as a composition of smooth surface patches which meet at curved

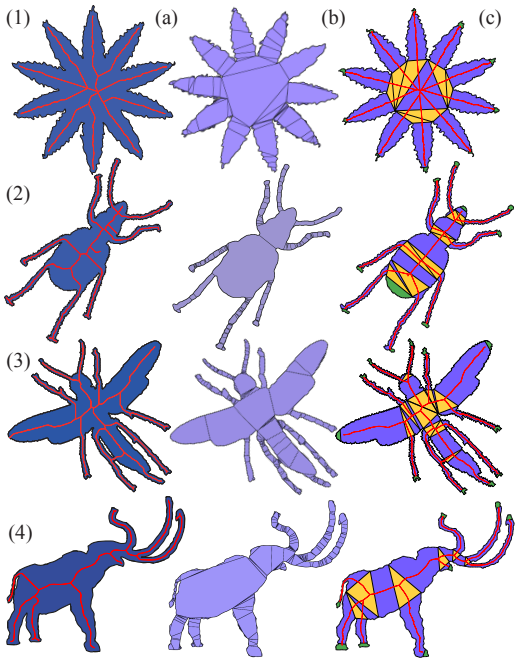


Figure 3: Comparison of the presented method (c) with the approach of Beeson et al. (a) and Lien et al. (b).

 ISBN 978-80-86943-76-3

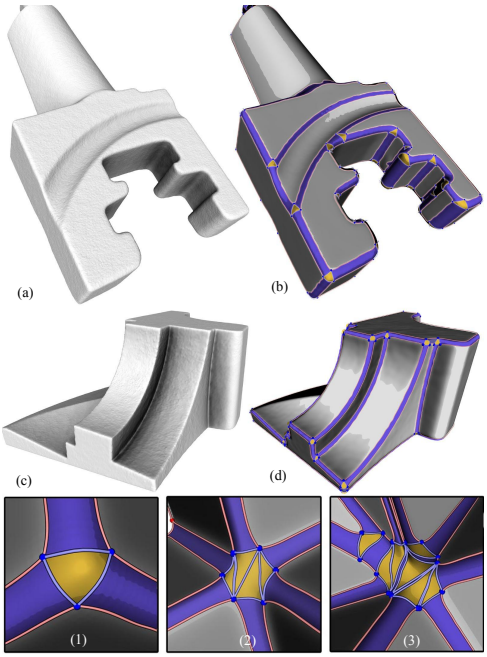


Figure 4: The decomposition is applied in digital shape reconstruction.

regions. The approach presented by Wekel et al. is used to segment and classify the surface into smooth and critical regions [TW13]. These regions represent a skeleton that captures the topology of a triangulated surface. In order to describe the model by typical CAD entities such as vertices, curves and parametric surfaces, it is important to decompose the polyhedral surface into a set of reasonable components. The reconstructed CAD models can be seen in (b) and (d) of Figure 4. Smooth surface patches are described by trimmed b-spline surfaces (grey). Using the presented algorithm, the blending regions are decomposed into junctions (yellow) and linear segments (purple). The presented method can be directly transferred to the domain of 2D manifolds in $\mathbb{R}^3$. As described in Section 3, each resulting part is characterized by a unique and compact structure which enables a straight forward representation in common CAD systems. The images (1), (2) and (3) in Figure 4 demonstrate, how the Voronoi-based approach allows to uniquely decompose even complex junctions into a set of simple, triangular surfaces and linear parts.

5 REFERENCES

[Aic96] Aurenhammer F. Alberts D. Gärtner B. Aichholzer, O. A novel type of skeleton for polygons. In A. Salomaa H. Maurer, C. Calude, editor, *J.UCS The Journal of Universal Computer Science*, pages 752–761. Springer Berlin Heidelberg, 1996.

[Bra92] Algazi R. Brandt, J. Continuous skeleton computation by voronoi diagram. *CVGIP: Image Understanding*, 55(3):329 – 338, 1992.

[CV12] B. Weber J. Halatsch D. Aliaga C. Vanegas, T. Kelly. Procedural generation of parcels in urban modeling. *Comp. Graph. Forum*, 31(2pt3):681–690, May 2012.

[HL10] L. Latecki H. Liu, W. Liu. Convex shape decomposition. In *CVPR, 2010 IEEE Conference on*, pages 97 –104, june 2010.

[JL06a] N. Amato J. Lien. Approximate convex decomposition of polygons. *Comput. Geom. Theory Appl.*, 35(1):100–123, August 2006.

[JL06b] N. Amato J. Lien, J. Keyser. Simultaneous shape decomposition and skeletonization. In *Proc. of the 2006 ACM symposium on Solid and physical modeling*, SPM '06, pages 219–228, New York, NY, USA, 2006. ACM.

[NA01] S. Choi R. Kolluri N. Amenta, Nina. The power crust. In *Proc. of the sixth ACM symposium on Solid modeling and applications*, SMA '01, pages 249–266, New York, NY, USA, 2001. ACM.

[PB05] B. Kuipers P. Beeson, N. Jong. Towards autonomous topological place detection using the extended voronoi graph. In *IEEE ICRA05*, pages 4373–4379, 2005.

[RO95] O. Kübler R. Ogniewicz. Hierarchic voronoi skeletons. *Pattern Recognition*, 28(3):343 – 359, 1995.

[Ros00] P. Rosin. Shape partitioning by convexity. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, 30(2):202 –210, mar 2000.

[RS04] A. Telea R. Strzodka. Generalized distance transforms and skeletons in graphics hardware. In *VisSym*, pages 221–230, 2004.

[SM12] J. Lang S. Montero. Smi 2012: Full skeleton pruning by contour approximation and the integer medial axis transform. *Comput. Graph.*, 36(5):477–487, August 2012.

[TW13] O. Hellwich T. Wekel. Voronoi-based extraction of a feature skeleton from noisy triangulated surfaces. In *ACCV2012*, volume 7725 of *Lecture Notes in Computer Science*, pages 108–119. Springer Berlin Heidelberg, 2013.

[TZ01] F. Preteux T. Zaharia. 3d-shape-based retrieval within the mpeg-7 framework. pages 133–145, 2001.

[WA12] B. Jüttler W. Aigner, F. Aurenhammer. On triangulation axes of polygons. *European Workshop on Computational Geometry*, 28:125–128, 2012.

[XB07] W. Liu X. Bai, J. Latecki. Skeleton pruning by contour partitioning with discrete curve evolution. *IEEE PAMI*, 29:449–462, 2007.

Refined Flattening Calculation of Hot Air Baloon Shape Surface Tiles

Radek Kubicek
Faculty of Information Technology, BUT
Bozetechova 1/2
61266, Brno, Czech Republic
ikubicek@fit.vutbr.cz

Pavel Zemcik
Faculty of Information Technology, BUT
Bozetechova 1/2
61266, Brno, Czech Republic
zemcik@fit.vutbr.cz

ABSTRACT

Design of hot air balloons can benefit from computer graphics in combination with numerical methods. The balloons are usually designed using modern CAD tools where their shape is modelled in 3D space. However, the design of the 3D shape is just a beginning of a complex production process. Flattening of the designed tiles and their preparation for a fabric cut is critical for a final quality of the design. Because of a relatively small number of companies producing the hot air balloons, only a limited number of very expensive useful tools exists. This paper addresses the issues of an automatic cut design calculation based on the 3D shape and physical properties of the fabric, using advanced numerical methods combined with existing flattening algorithms.

Keywords

surface flattening, cut design, energy model, particle system, numerical system, differential equations, hot air balloon design

1 INTRODUCTION

Many products of complex 3D shapes are being manufactured from materials, such as fabrics, tins, etc., that are being produced as flat ones. Therefore, preparation of the manufacturing process – "surface flattening" is an important process in many applications, in our case manufacturing of hot air balloons. The hot air balloon envelope, its design, shape, and the graphics applied on are the main distinctive attributes of the balloons and they are the main cause for their complex and difficult manufacturing. The envelope is constructed from gores made of polyamide or polyester fabric, reinforced with sewn-in load tape. The gores, which extend from the base of the envelope to the crown, comprise of a number of smaller tiles connected together with the seams. Balloons' envelopes can be divided into various types with similar shape and differences only in a graphics design and special shapes, which are unique. The envelope is important as it not only defines the shape and appearance of the hot air balloons but also their flight characteristics.

The envelope shape is typically constructed in a 3D/CAD software. However, what the balloon industry needs is not only a 3D simulation of the shape but rather fine design of the "flat" 2D patterns which can be used in the manufacturing process. Flattening is often not too difficult in case of basic shapes; however, special balloon types, which can have very complex shape and whose envelope is difficult to flatten, became recently very popular. While the 3D model represents the inflated uptight shape of the balloon, after flattening to the cut design plane its shape and some parameters could be little different. In general, the most important parameters are the lengths of the separated tile edges, whole tile patch circumference, and shape of the patch.

This paper addresses an approach leading into a design of the suitable 2D shape from a given 3D surface by using a combination of flattening techniques to simulate the process of creating a 2D form. It also describes methods which are used during this process, an overview of its advantages and disadvantages and results comparison to some related works.

1.1 Related work

Research in surface flattening area has been active for many years already, therefore, many methods have been designed with various parameters and usability features. First known class of methods are the parametrization methods which are based on a bijective mapping between the original mesh and the planar mesh – the

 ISBN 978-80-86943-76-3

flattened surface. These methods use mostly numerical methods and linear equations system. In our work, we are not using this class of methods. An excelent survey of recent parametrization methods is given in [Wang05], see also the references therein.

Second class of methods is formed from the methods based on minimization of a strain-energy in the 2D flattened pattern. After the projection onto the plane, the length differences are measured and treated as a strain energy and some iterative method is then applied to minimize this strain energy. Several methods exist and differ in the way the energy minimization is performed; however, they have the energy minimization scheme applied on the 2D patterns in common. McCartney et al. [McCart99] proposed a flattening of a triangulated surface by minimizing the strain energy in the 2D pattern. Wang et al. [Wang02] improves this method by using a simple spring-mass system. More details about strain-energy minimization methods can be found in [Wang05].

Woven fabric related models can be considered another class of methods. Woven fabrics consist of a series of crossed vertical and horizontal threads. In [Aono94] three basic assumptions were stated; (1) the strength of the threads is usually very high, threads are inextensible, (2) thread segment between adjacent crossings is straight on the surface, and (3) no slippage occurs at a crossing during fabric deformation. These methods generally rely on physical properties of the fabric and try to approximate it; equidistant points are mapped on the original surface under predefined direction. The main problem of this class of methods is that the resulting plane pattern depends on the selected position and direction of the base line. More details about this class of methods can be found in [Wang05].

Our previous work [Kub10] was based on a simple surface planar projection followed by the woven fabric model application that finetuned this projection. This method depended heavily on the early surface projection and it inclined to incorrect results in case of complex or strongly curved surfaces. This paper proposes an improved flattening method based on a combination of the woven fabric model approximation algorithm described in [Wang05], the strain-energy minimization method used in [Wang02], and the energy releasing spring-mass system as a refinement part of the flattening process.

2 FLATENNING PROCESS

An input of the flattening process is the 3D design of the envelope surface divided into several tile meshes M_x. In our case, it is designed using Rhinoceros 3D software. This design has pre-defined locations of the seams and tiles and also the fabric yarns orientation. The main constraint, comparing to other similar tasks is that it is not possible to create any new seams neither insert any darts. It is necessary to flatten given tile as it is or detect if it can not be flattened within a given accuracy limits and it needs to be redesigned.

```
Find planar mesh approximation Γ of 3D input mesh M
while steps < max_steps do
    for node in Γ do
        CalculateForceAndDerivative()
    UpdateParticlesPositions()
    UpdateEnergy()
    if success metrics fulfilled then
        break
```

Algorithm 1: Pseudo-code of the fine tuning algorithm.

The proposed model assumes an orthotropic material, which is used where the elastic performance is sensitive to the grain direction with respect to two orthogonal axes: warp and weft. The actual input 3D surface representation is a polygon mesh; the quality of this underlying mesh is important to the success of the flattening process. Polygon mesh is created for each envelope tile, a quality of the tiling is critical for suitable result.

The flattening process itself consists of several steps applied sequentially. The first step is finding a tile mesh M planar mesh approximation Γ which is accurate as most as possible. This goal could be achieved in many various ways, we use a slightly modified woven mesh fitting algorithm described in [Wang05] in detail.

2.1 Fine tuning process

Obtained mesh grid Γ is just a tile shape approximation and it needs to be refined in a fine tuning process which is a next step of the flattening process. We propose a particles spring-mass system simulation in combination with numerical methods and Hook's law. Approximated mesh Γ is given as an input of this simulation, where mesh nodes represent masses and the links between them represent springs. In this paper, we use an energy model; planar triangular spring-mass system to obtain the refined flattening result, inspired by the [Wang02]. Thus, the process of flattening is a deformation process driven by the energy function of the spring-mass particle system.

To simulate the dynamics of mesh grid, we integrate the system of differential equations over a time. At each time step $t_{i+1} = t_i + \Delta t$ we sum all of the forces acting on each node. Force is calculated for every mesh spring using equations

$$\vec{F_x} = -(K_{weft} \cdot (l - l_0).x + K_d \cdot \Delta v.x \cdot \vec{x}) \cdot \vec{x} \qquad (1)$$

$$\vec{F_y} = -(K_{warp} \cdot (l - l_0).y + K_d \cdot \Delta v.y \cdot \vec{y}) \cdot \vec{y}, \qquad (2)$$

where l_0 is a spring rest length, K_{weft}, respectively K_{warp} are weft and warp direction parameters calculated from physical fabric properties in combination

 ISBN 978-80-86943-76-3

with length difference. A dumping factor K_d takes into account in the velocity calculation. Standard Newtonian equations of motion and a midpoint method are used to advance the current velocity and position over the each time step. This method has advantages in a faster convergency and its stability. With using a small derivative step, we can get reliable method which produces precise results. We will focus in our future work on an adaptive derivative step which can make the iterative process more precisely. This solution system needs to be stable and it should not let the grid to deform or the calculation stuck. However, if some spring has a length error out of the bounds, a system may become unstable and such behaviour needs to be detected and corrected immediately.

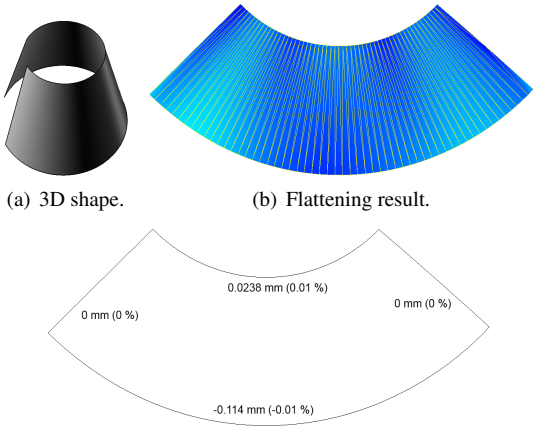
2.2 Definition of a success metric

Three accuracy criteria are used to evaluate the accuracy of the flattening process resulting surface. All of them are used as a termination criteria for the flattening finalization procedure. **Area accuracy.** Should not exceed 1 % difference between the original surface area and the flattening process resulting mesh area. However, this also depends on a complexity of the surface, more complex surfaces may have bigger are difference due to less precise result. **Shape accuracy.** We can distinguish between two types of the shape accuracy. The accuracy of each edge length and the difference between the total edge length on the M and on the Γ. A required accuracy is the difference of up to 1 mm over 1 m of length (0.1 %). However, this could be violated in case the length difference of a matching edge of an adjacent tile and overpassed edge already fulfil required accuracy. **Energy accuracy.** The most important accuracy criteria is the mass-spring system energy minimization meaning, in fact, proper overall shape. During each step of iteration, the mass-spring system energy is calculated. This value is then compared with the one of the previous step of iteration. If the energy stops decreasing or the difference between last two steps is smaller than selected treshold value of an allowed minimal energy step, the iteration process is stopped.

Area and shape accuracies are used just only as an additional metric of success and they are checked right after a completion of the flattening process. Only if the energy minimization criteria is fulfilled, these other two metrics are checked. The area and shape metrics are used to simplify a result evaluation, minimization itself would be very difficult to implement due to its complexity. The main decision is whether the tile is manufacturable and if so, result is the proper shape of the cut. If not, the balloon design should be changed.

3 EXPERIMENTS AND RESULTS

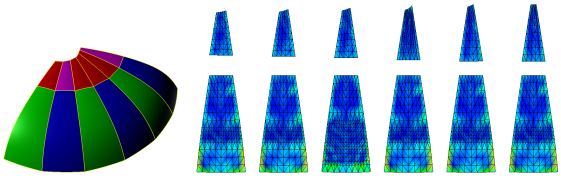
Figures 1, 2 and Table 1 show results of performed experiments with tiles which are close to developable surfaces or easy to flatten. Figures 3, 4 and Table 2 show results of experiments with complex and not developable surfaces. The order of the figures top to down: 3D shapes of panels, flattening results, and a comparison of the edges (or adjacent edges if possible) accuracy. Red colour in the flattening results image means the highest error out of acceptable bounds, blue means the smallest or no error. For tables, values in italics are stated for comparison with the other methods; however, the input data is not the same, it is only similar, so the results are not very well comparable. Minimal energy criterium was set to the same threshold value for each experiment, it is not necessary to state it.



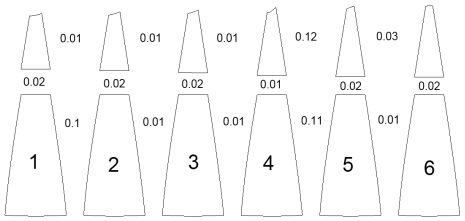
(a) 3D shape. (b) Flattening result.

(c) Edges accuracy (in *mm*).

Figure 1: Developable surface - a cone.



(a) 3D shapes. (b) Flattening results.



(c) Adjacent edges accuracy (in *mm*).

Figure 2: Examples of well developable surfaces.

The performed experiments prove that the implemented fine tuning algorithm is very well usable for the various shapes of tiles and it is the mostly limited by an application internal meshing process. Currently, the physical properties of the fabric are not still fully implemented in the flattening algorithm; we expect result improvements of this aspect in the future work.

 ISBN 978-80-86943-76-3

Tile	Area err. (%)	Shape err. (%)
Cone	**0.07**	**0.0055**
Cone [Wang02]	*0.092*	*0.174*
Cone [Wang05]	*0.25*	*0.05*
Figure 2 (bottom 1, 5, 6)	0.00	-0.01
Figure 2 (bottom 2, 3, 4)	-0.01	0.04
Figure 2 (top 5)	0.00	0.03

Table 1: Flattening results of quite well developable surfaces.

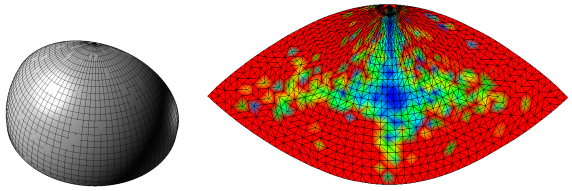
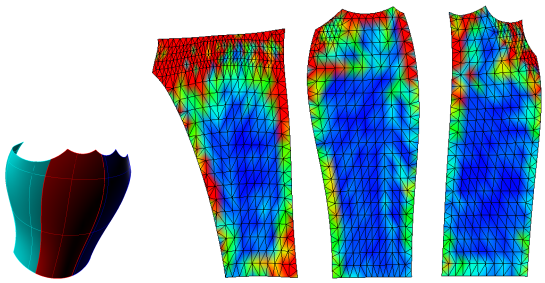
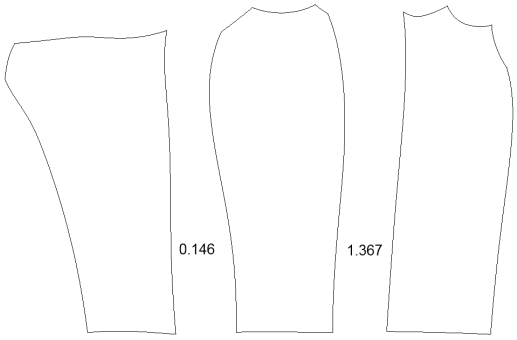


Figure 3: Example of quarter of sphere flattening. The result is unacceptable and the tile must be further subdivided.



(a) 3D shapes. (b) Flattening results.



(c) Adjacent edges accuracy (in *mm*).

Figure 4: Examples of non easily developable complex surfaces.

Tile	Area err. (%)	Shape err. (%)
Quarter of sphere	1.93	14.01
Figure 3 left	0.87	0.19
Figure 3 middle	0.08	0.17
Figure 3 right	0.18	0.30
Garment [Wang05]	*0.30*	*1.79*
Garment [Wang05]	*0.25*	*0.05*

Table 2: Flattening results of non easily developable complex surfaces.

4 CONCLUSION

The presented work focuses automatic flattening algorithm of hot air balloon envelopes, mostly its flatten shape fine tuning. We implemened a numerical model useful for solving the given flattening problem – a combination of a woven mesh fitting, particle system solving, and numerical methods application, and we also defined the success metric which the results of flattening problem needs to meet. The input of the presented algorithm is the previously designed 3D envelope model divided into several tiles and the result is the cut design of flattened tiles. After we apply the presented algorithm on all of the balloon envelope tiles, we receive the whole envelope cut design, where all problems caused by wrong envelope design and/or tile division can be simply detected.

The results obtained by the automated process are promising. They generally differ from the ones obtained actually "by hand" and they were evaluated as having the same or better quality. Also, they are much faster and easier to obtain. The proposed algorithm is currently used in the manufacturing process and the results are being compared to the previous manufacturing processes. In our future work, we will focus on a better handling of the physical properties of fabric in proposed algorithm and on the more precision of the refinement results mainly.

5 ACKNOWLEDGMENTS

We thank to Kubicek Balloons company for technical advisement and providing data; thanks to their courtesy we were able to use their balloon designs in our research. This work has been supported by the research projects TE01010415 (V3C – Visual Computing Competence Center) and IT4Innovations (MSMT, ED1.1.00/02.0070).

6 REFERENCES

[Aono94] Aono M., Breen D.E., Wozny M.J. Fitting a woven-cloth model to a curved surface: mapping algorithms. Computer-Aided Design 26, pp.278-292, 1994.

[Kub10] Kubicek, R., and Zemcik, P. Balloon Cut Layout Based on 3D Balloon Model, in Proceedings of the IADIS CGVCVIP 2010, Freiburg im Breissgau, DE, IADIS, pp.57-64, 2010.

[McCart99] McCartney J, Hinds B.K., Seow B.L. The flattening of triangulated surfaces incorporating darts and gussets. Computer-Aided Design 31, pp.249-260, 1999.

[Wang02] Wang, C.C.L., Smith, S.S-F., and Yuen, M.M.F. Surface flattening based on energy model. Computer-Aided Design 37, pp.823-833, 2002.

[Wang05] Wang, C.C.L., Tang, K., and Yeung, B.M.L. Freeform surface flattening based on fitting a woven mesh model. Computer-Aided Design 37, pp.799-814, 2005.

Comparative Visual Aesthetics in Synesthetic Structures

Adrian Johnson SK Semwal

Department of Computer Science
University of Colorado
Colorado Springs, CO, USA
adrian42@gmail.com

ssemwal@uccs.edu

ABSTRACT

We describe a means of co-evolving parameters for procedural animation generation. Several experiments were completed in order to find an aesthetically pleasing animation mechanism, which could be modulated by music and have its parameters evolved over time. The selected music visualization is a set of vector fields for the purpose of moving particles whose paths create visual interest. To cause emergence of pleasing patterns in particle behaviour, several metrics were incorporated into the fitness function. To reflect the spatial characteristics of the vector fields a crossover operator was developed. The most critical aspect in creating a vector field which animates particle paths over time is that the vector field not allow the particles to become static or too tightly clustered, as either indicates the particles will no longer be circulating in interesting ways. We are interested in encouraging certain behaviours of the particles without doing so explicitly so that a local interaction approach allow for many aesthetically pleasing solutions to be reached, and rendered. This complex systems approach continues to work as interesting patterns emerge. We provide several examples with later generations creating more aesthetically complex renderings. Comparative visual aesthetics as a measure of such visual refinement is introduced. If pleasing patterns can be made implicitly with the vector fields in this specific audio-visual composition, it stands to reason that parameters for other visualizations may be evolved as well and this work serves as an argument for computer assisted comparative aesthetic refinement.

Keywords

Particle animation driven by music; vector field rendering; local interactions creating global effect

1. INTRODUCTION

One purpose of this paper is to explore computer assisted generation of audio-visual art. The question of what is interesting is usually based on the observer [Ols05]. Just as impressionist paintings are pleasing from a great distance when the artist's dithering and the interpolation by the human visual system and have an interplay; those same paintings have fascinating brush strokes when viewed up close. The link between music consumption and the emotional state of a person is not entirely understood, but a connection between the two has been proven many times [Cam01]. One successful examples of interactive is the pleasing works through an understanding of aesthetics [Kur00].

2. VISUAL AESTHETICS

Computational aesthetics incorporates computer science, neuroscience, and visual art. It is known that symmetry plays a part in our determination of what we consider beautiful [Tyl00]. Synesthesia occurs when several phenomena are perceived in unison [Joh07, Hub05].

When a collection of entities move together, using rules which govern individuals within the group, swarming behavior results and has been modeled computationally [Tho05] and cellular automata earlier [Joh02; Cho00; Gol89; Wol02; Sim94]. (Figure 1).

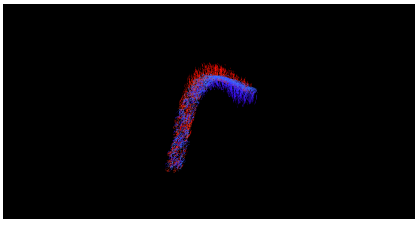


Figure 1: Structures imposed on the swarm through initial position and vector field edges.

Our second example is a frames of flowing river which provides rich textures found in the water. We experimented with accentuating theses characteristics – either strong lines or indistinct splashes of texture – by using the images as height maps (Figure 2). Each

pixel corresponds to a color and z component which was rendered using mesh using openGL™.

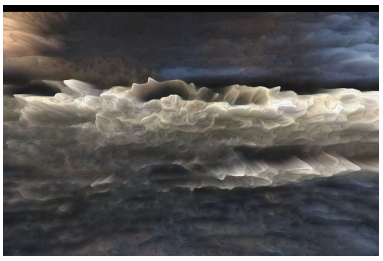


Figure 2: Height Map interpretation of river video frame.

We ran a Gaussian filter on the video frame to tune the amount of noise and the sharpness of the resultant geometry (Figure 3), including text and a grid of squares.

Study of Figure 1-3 provides us the opportunity to define the *comparative visual aesthetics* which is defined as a measure of visual-interest an image or a piece of art generates when compared with another. Figure 1 does not have the same visual interest as in Figures 2 and 3. Figure 2 and 3 create similar visual interest, and so the measure of their visual interest is similar to us. This measure is both subjective and relative and is proportional to time we want to gaze, admire, or look at the piece of art or image. As visual aesthetics is based on perception, large variation of measures can occur. Yet it is much easier to be consistent when one image or art piece is compared side by side with another for developing sense of relative measures instead of absolute measures. Comparative visual aesthetics provides one relative and subjective measure. Computational measure of complexity of an art pieces is an open area of research. Our measurement of aesthetic complexity is along several lines of current research such as the work by Itti, Dhavale and Pighin [Itt04] and Santella [San05]. Yevin's [Yev02] notion that the criticality of the art is related to patterns and rhythms is of major interest for us in developing our concept of *comparative* visual aesthetics.

Figure 3: Addition of graphics elements to the height map.

3. SYNESTHETIC STRUCTURES

As generations of work can be visually compared to each other and only visually interesting pieces survive, one of the thoughts we have pursued is that we would definitely need a generative-system, which inherently contains, and which can match the desired aesthetic-complexity. Since we had used music as an input device in our earlier work [Joh04, Joh07], we decided that music could also drive the generation of such synesthetic shapes. So synesthetic shapes are generated by varying the properties of particles or cylinders based on the analysis of the music. For example, thickness or radius of cylindrical shape could be dependent upon the presence and amplitude of certain range of frequencies. The trace of the particles could be thought of as vector fields being created at many levels of resolution and in some ways being affected by a weighted combination of nearby vectors. With that in mind we added a few more stochastic mutations of the vector field resulting in synesthetic structures as rendering of these vector fields as 2D ribbons [Joh07]. The 2D ribbons' paths were created by using square tubes which followed the particle paths created by vector fields. We formed these tubes by finding the normal of the most recent segment of path and a vector from the most recent point to the origin. An orthogonal vector was then found using the first normal and the vector from the most recent point to the origin and taking their cross product. Those two vectors were reversed to determine the corresponding points on the opposite side of the path. Triangles were used to create a tube (Figure 4) between determined set of points. We used the kick drum events to drive one particle-swarm (the grey particles in Figure 5) and the snare drum events to drive another, ignoring other events in the MIDI sequence. Because the snare events were more prevalent than kick drum events, the kick drum events were weighted higher to cause both particle-swarms to contribute structure to the swarm based sculpture almost equally. When the particles intersected each other we caused the growth to pause, and the path would be rendered white to show the particle was no longer contributing to the structure. One can see the lines in several states of construction (Figure 5), from barely growing to many particles intersecting. As shown in Figure 6, the reddish swarm becomes more orange as the particles age, with only the most recent path segments being red. The grey swarm becomes predominantly white as its members are driven to an edge within the vector field's cube of influence (Figure 6).

Particles from the same swarm tend to weave concentric patterns and slight intersection in tube geometry is visually interesting when rendered, it looks like fluid wires melting together and then diverging again. Instead of tubes with equal width and thickness, we caused the thickness to be somewhat less than the width, giving us something of a ribbon effect. This made the paths' position relative to the origin very interesting, causing the ribbon to twist during turns. This twisting is due to the use of the cross product of the vector from the particle to

 ISBN 978-80-86943-76-3

the origin being used to calculate the initial normal used in constructing the tube geometry. We then had the particles create the tube geometry using not the origin for the initial normal, but an animated point near the origin. The point is at the origin in the X and Y dimensions, but varies with the MIDI impulse along the Z dimension. This point was different for each particle-swarm since each particle-swarm is driven by different MIDI events, causing the swarms to appear to *breathe and pulsate* in rhythmic harmony. Lastly, to have some interplay between swarms, each swarm is affected by 100% of its vector field's strength and 10% of each of the other vector fields. This manifests in patterns of similarity between particle-swarms, as though the swarms were slightly chasing each other. The most interesting part of Figure 7 is that, comparatively, we feel that it has more visual aesthetic-complexity or visual interest then images in Figure 6.

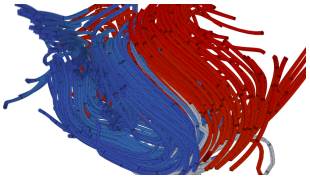


Figure 4: Depth of field effects with geometry overlay.

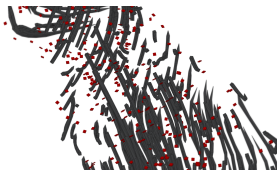


Figure 5: Early stages of synesthetic sculpture.

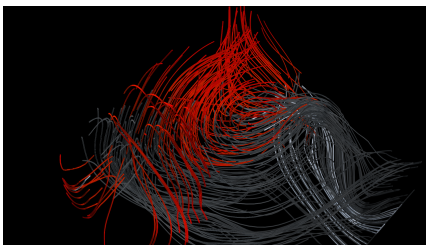


Figure 6: A frame showing complexity growth in synesthetic structures simulation.

Genetic Algorithm

Figure 8 depicts an ending condition of an individual. The three swarms are grey, yellow-orange, and red, with particles that have collided with vector field borders rendered in a lighter version of those colors in order for the user to observe the relative success of the swarms. In these evolved sequences, the red particles are moved by the kick drum, the grey particles are linked to the snare drum, and the orange particles are moved by the high hat in the drum sequence. We experimented with different variations on the crossover, animation duration, and other aspects [Itt04], and started with the same initial particle positions and vector field configurations across all executions using (GA) operators [Gol89](Figure 10, 11).

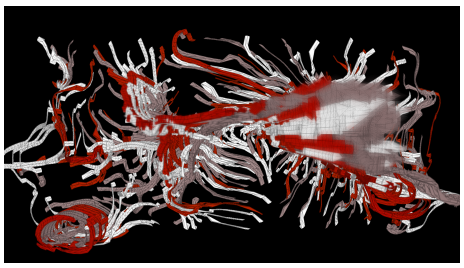


Figure 7: Visually interesting concentric paths shown on lower left and variation creates large aesthetic complexity compared to Figure 6.

Figure 8: Three swarms corresponding to drum components.

Implementation Details

SwarmGA, implemented using Java OpenGL bindings (JOGL) is explained in [Joh07]. SwarmsGA runs in batch, consisting of several, typically 100 generations, and takes on average six hours to complete on a 2.5 Ghz G5 computer. Fifty swarms over 100 generations were tracked, and displayed.

4. RESULTS

We will present the results of several configurations of the system, with revisions to the genetic operators interspersed. As a preface to the per-generation illustrations, in Figure 9 one may see a general example of an unfit system (generation 1) and a fit system (generation 100).

Emergent Behavior: An interesting thing to note is the emergence of different curvatures to prevent the particles from meeting boundaries in Figure 9. This took the form of switchbacks and vortices, which are always interesting shapes. The concentricity created patterns among members of the same swarm, and the slight weighting of the other vector fields to each

 ISBN 978-80-86943-76-3

swarm in addition to its vector field caused some interplay and co-evolution between vector fields.

Computational Visual Aesthetics: While we predicted some trends such as vortices would occur we were surprised at how much more appealing later generations appeared. The visual aesthetic value of generation 100 *is* more than that of generation 1 which is based on increase in folding patterns. In [Joh07], several swarms were traced and generally higher generation showed more comparative visual aesthetics, and thus better fit as vortices were encouraged during evolution and edges were discouraged from one generation to another. Generally, we can say that Figure 7 has higher comparative visual aesthetic value than Figure 6 due to color variation as well as variety. As we learn more about what contributes to beauty from a computational aesthetics point of view we will be able to expand upon the existing fitness function. When animations were created from the high fitness individuals, the concentric paths were especially appealing to watch being constructed. The degree of swarming and total motion in the scene was greater in highly fit individuals, contributing to the quality of the result a shown over several generations in [Joh07]. In a sense the later generations exhibited a greater aesthetic value, beauty.

5. CONCLUSION

The role of computers in the arts encourages comparisons based on aesthetic complexity. We proposed the concept of comparative visual aesthetic in this paper. Our work on musically driven synesthetic shapes and structures led to an implementation where we were able to compare the visual aesthetics across several generations and show the results. Fitness scores as a computational measure of visual aesthetics is an important area of research, which needs to be further investigated. Although correlation is expected, a direct relationship between visual aesthetics and fitness scores is not proven as different rendering styles can and do lead to different visual aesthetics. We would also like to study speciation in relation to this problem. A sharing function would enable us to refine a system with multiple potential swarms that evaluate with good fitness. Further exploration of the varying crossover operators would help to avoid destroying vector field features that contribute to the fitness. To improve execution speed, graphics processing unit (GPU) implementation will benefit the execution time. Rather than multi-thread the program, we could run an instance of the program per CPU core and should see dramatic improvement in execution speed. Finally, more evaluation of comparative computational aesthetics would be necessary to develop this field further.

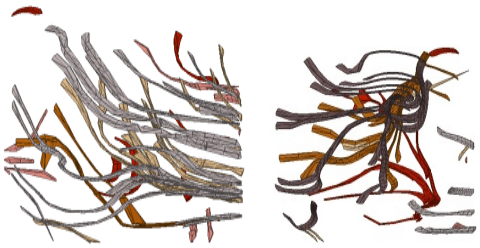


Figure 9: Substantial folding and complexity improvement in fitness between generation 1 (above) and generation 100 (below).

6. REFERENCES

[Cam01] Campbell, Don (2001). *The Mozart Effect*. New York City, New York, Harper Paperbacks.

[Cho00] Chouchoulas, Orestes (2000). Shape Evolution: An Algorithmic Method for Conceptual Architectural Design Combining Shape Grammars and Genetic Algorithms. fufurasu.org/research/aid02abstract.pdf .

[Gol89] Goldberg, D.E. (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning*. USA, Addison Wesley.

[Itt04] Itti, L., Dhavale, N., and Pighin, F. 2004. New realistic avatar and head animation using a neurobiological model of visual attention, Proceedings of SPIE 48th Annual International Symposium on Optical Science and Technology.

[Joh02] Johnson, Colin G., Romero Cardalda, Juan J. (2002). Genetic Algorithms in Visual Art and Music. *Leonardo,* Volume 25, Number 4, August 2002, p 364.

[Joh06] Johnson, Adrian (2006). Complexity and Music Generation, *GECCO 2006 Proceedings, Graduate Student Workshop*. Seattle, Washington, July 2006

[Joh07] Johnson, Adrian. Aesthetic Emergence in Synesthetic Structures, MS Thesis Advisor: SK Semwal, Department of Computer Science, University of Colorado, Colorado Springs, pp. 1-67 (2007).

[Kur00] Kurzweil, Ray (2000). *The Age of Spiritual Machines*. New York City, New York, Penguin.

[Ols05] Olsen, Sven, Maxwell, Bruce A., Gooch, Bruce (2005). Interactive Vector Fields for Painterly Rendering, *Graphics Interface 2005 Proceedings*. Canada, The Canadian Human-Computer Communications Society, pp. 241-247.

[San05] Santella, A. 2005. The Art of Seeing: Visual Perception in Design and Evaluation of non-Photorealistic Rendering, PhD thesis, Rutgers. www.research.rutgers.edu/~asantell/thesis.pdf.

[Sim94] Sims, K. 1994. Evolving Virtual Creatures, Proceedings of ACM SIGGRAPH, 14-22

[Tho05] Thomson, Elizabeth (2005). "Science becomes art in 'Weird Fields,'" The MIT Press.

[Tyl00] Tyler, Christopher W (2000). The Human Expression of Symmetry: Art and Neuroscience. Web-reference at: www.ski.org.

[Wol02] Wolfram, S. (2002). *A New Kind of Science*. Canada, Wolfram Media.

[Yev02] Yevin, I. 2002. Criticality of the Brain and Criticality of the Art, Proceedings of the Fourth International Conference on Complex Systems, Book Chapter in Unifying Themes in Complex Systems IV. Editors AA Minai and Yaneer Bar-Yam, Springer-Verlag, pp. 168-174 (2010).

A Visualisation of the Results of a Thermoforming Process Simulation in the Plastics Industry

Michal Gerža
Tomas Bata University in Zlín
Faculty of Applied Informatics
Nad Stráněmi 4511
76005, Zlín, Czech Republic
michal.gerza@email.cz

Pavel Pokorný
Tomas Bata University in Zlín
Faculty of Applied Informatics
Nad Stráněmi 4511
76005, Zlín, Czech Republic
pokorny@fai.utb.cz

ABSTRACT

This paper presents the SIMViewer computer program allowing users to visualise the results from the T-SIM simulation software made by Accuform in the field of plastics thermoforming. The main feature is to visualise particular thermoforming process steps of a plastic sheet in a three-dimensional space including all geometric parts of the defined model like a forming die and optional plunger. SIMViewer provides a projection of scalar quantities that characterize the thermoformed region. It is mainly focused on presenting the thickness distribution, thermoforming temperature, stress, strain and other scalars applied on the plastic sheet. This application was developed in C++ programming language, the graphical area was implemented by the OpenGL library and the user interface was created by using the MFC application framework.

Keywords

visualisation, simulation, geometry, framework, thermoforming, plastic

1. INTRODUCTION

In recent years, the industrial world has expanded into computer simulations which determine the direction of product development in terms of suitable materials usage and manufacturing technologies. Simulations are important especially today, during economic instability, when companies are forced to reduce their production costs in order to compete in the market. Here, simulations come either in the form of mathematical models based on an input set of data and characteristics and they can generate the behavior of an object in time and space, or optimize its geometric parameters for effective use. These models are currently very sophisticated based on many types of algorithms, so the simulated solutions often approach reality. This is also true for the Plastics Industry. The mathematical modeling results allow teams of skilled engineers and technologists to diagnose eventual product weak points and, for example, to optimize the consumption of material used during the product manufacturing process.

The aim was to create a 3D visualisation program that would be able to transfer data interpretation of the modeling results into a clearer graphical view. Technologists can then see the optimal plastic products wreaked by the thermoforming process. It is frequently used for the comparison and verification of final data results of different case studies.

2. THE PLASTICS PROCESSING

The plastic products manufacturing requires specific technological processes, specific machines and tools in order to achieve products like a plate, cup, toy or electrical device covering and fuel tank. It depends primarily on the complexity and shape of a final product, on used material and influences which affect the product in the real life, such as weather, chemical and physical laws. Plastics, generally called polymers, are processed by the six basic technologies like the thermoforming, blow molding or extrusion, injection, pressing and rolling.

This paper is going to focus on the thermoforming technology only by reason of its diversity in terms of applicable methods for processing of various plastic materials. The reason of preferring this technology was a presentation of the SIMViewer application developed for the visualisation of final results from a computer simulation of the thermoforming process.

2.1. The Thermoforming Technology

Thermoforming is a specific process in which a flat thermoplastic sheet is heated and deformed into the desired shape. This technology is used up for a packaging and to fabricate large items like furniture, contoured skylights and automotive parts. [1]

This processing technology consists of two main steps, heating and forming. Three basic methods are standardly used up like the vacuum, pressure and mechanical thermoforming process.

 ISBN 978-80-86943-76-3

3. A COMPUTER SIMULATION OF THE THERMOFORMING PROCESS

Computer simulations have become a useful part of mathematical modeling of many natural systems, for example, in physics, chemistry, biology, and also in the process of engineering new technology, to gain insight into the operation of those specific systems. Some simulation methods have been applied in the field of thermoforming process as well.

3.1. The Finite Element Method [2]

Generally accepted method for a thermoforming simulation is Finite Element Method (FEM). The main goal of FEM in a thermoforming simulation is thickness prediction. Approximate wall thickness values for complex mold geometries are obtained by assuming that the actual shape of the mold surface is approximated by a simple geometry or series of geometries. FEM is a scheme for determining the wall thickness of well-behaved stretched membranes. The general scheme is to overlay the surface to be stressed with a grid or network (see Fig. 1).

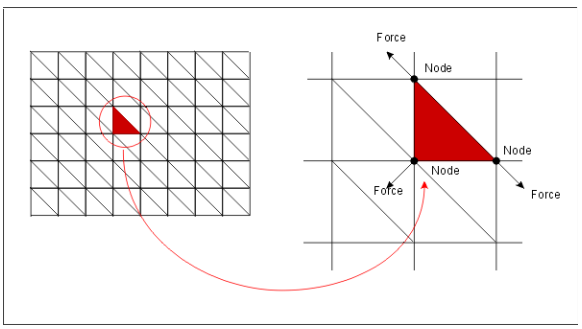


Figure 1. The FEM grid, elements and nodes

The grid pattern is usually triangular. Depending on the sophistication of the model, the elements are two-dimensional (2D) with X and Y coordinates but no thickness, or they are three-dimensional (3D), with X and Y coordinates and a finite thickness. It is thought that 3D elements are necessary when there may be heavy-gage sheet bending resistance over sharp edges or when shear is expected. For most applications, 2D or thin membrane elements are acceptable. The elements are connected via nodes. The connected elements form a discrete surface or mesh that replicates the actual continuous surface. When forces are applied to the simulated surface, the elements distort, with the extent of distortion determined by balancing the forces and moments at each intersection node. The relationship between applied force and resulting strain is called the constitutive equation of state of the material. Many thousands of such elements are usually needed to faithfully simulate structure response to applied load. Since many equations are needed for each node and three or more nodes are on each element, thousands of equations must be solved simultaneously to affect a solution. As a result the artificial computer time step must be very small to minimize error generation during each iterative calculation.

3.2. The Process Optimization [2]

Simulation approach is increasingly being used for understanding and optimizing polymer processes. Its advantages are that the whole process can be set up on a low cost computer, and all of the variables can be changed at will to study their effects. This saves time, machinery investment, costly raw materials, and operation expenses. In addition, simulation can allow a visualisation of effect which cannot be seen or measured in the process. When investigating the thickness distribution of a plate we can only see the starting position and the end result. With simulation, the operation can be studied and understood.

The objectives for using CAE tools have changed dramatically over the years. Their preventive use in the design and development phase is now called for rather than error diagnosis on finished products. This perspective can be described as the virtual prototyping. The decision-making aid is no longer an actual product; in its place, improvements and risk calculations can be realized on virtual variants.

If one looks in detail at the design or outline of a product, one always faces an optimization task. For example, shape and material must fulfill certain criteria relating to product characteristics and manufacturability. However, also functional with economic objectives must be met in the best possible way. Quality levels usually serve here as target objectives. In most cases, they are given verbally and might, to name but a few, include the following:

- *Low distortion characteristics*
- *Minimal component weight*
- *Low production costs*

Conditions must be satisfied that not only relate to the product itself, but also to its manufacturability and the related manufacturing process.

3.3. The Optimization Techniques [2]

Various optimization techniques are used to achieve satisfactory results as follows below:

1) *Material optimization* - This promising way of optimization is development of special materials to be used in the thermoforming industry. It is becoming more evident that resins exhibiting strain hardening during elongation are more

suitable for deep draw thermoforming. These materials indicate a kind of self-healing effect beneficial for the homogeneity of deformation.

2) *Optimization of initial temperature or thickness profile* - Modification of initial temperature or thickness profile on a sheet is often used when producing large formed parts.

3) *Plug assisted forming optimization* - In plug assisted forming, using different final heights, speeds, sizes and shapes of plugs leads to many different final material thickness distributions.

4) *Material of tools* - Friction between tools and sheet plays also an important role in the process, again mainly in plug assisted forming. It is clear that when the sheet almost totally slides on the surface of a plug, the wall thickness of material at the top of the applied plug is low.

5) *Optimization of pressure/vacuum level/rate of change* - using the rate of change and level of vacuum/pressure plays also a significant role in the process. As can be expected, the higher is the speed of vacuum or pressure, the higher stresses are generated in material and it has an impact on the final thickness distribution.

6) *Optimization using sagging* - The effect of sagging can be used also to influence the final quality of a product. When a sheet sags, the thickness profile on it is quite uniform compared with the profile obtained using pre-blow.

3.4. A Math View of the Optimization [2]

Any optimization task must contain a determination of set of independent variables and usually also includes conditions characterizing allowed values of the independent variables. The conditions are called task limitations. Needed part of an optimization task is a scalar quality measure (cost function). Its values are given unambiguously by values of independent variables. Solution of an optimization task is such an allowed vector of values of independent variables, for which the cost function has an optimal value. The optimal value of the cost function is usually its minimum or maximum in an allowed range of independent variables. In the case of optimization of complicated processes such is thermoforming, the simulation of the process is quite time-consuming. As a result, it is hard to implement generally well described optimization techniques requiring many calculations of the cost function.

4. THE VISUALISATION SOFTWARE

This paper focuses on the SIMViewer visualisation software by which we are able to visualise and analyze output data results produced by the T-SIM simulation software in the field of thermoforming.

4.1. The SIMViewer application

The SIMViewer application was fully coded in C++ programming language. The specific libraries of MFC (Microsoft Foundation Classes) and OpenGL were implemented for the application framework and advanced graphic interface. [3] This application has been built, compiled and optimized for Microsoft Windows XP/Vista/7/8 - 32bit version.

The graphical user interface (GUI) of SIMViewer includes the plentiful palette of features which help users to comfortably control and edit the visualised model of plastic material product. Most of these features are accessed in the main menu. Some of them are also placed in the tool bars and have own shortcuts which simplifying the manipulation with displayed model components (see Fig. 2).

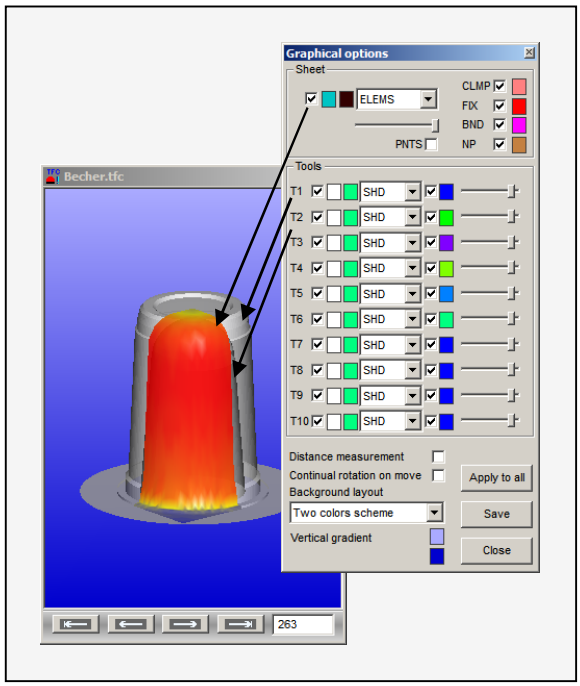


Figure 2. The Graphical options dialog box used for controlling of an active child window.

The SIMViewer application also defines input data files which are mostly in the binary form. The main data file has the TFC extension as a post-processing profile that defines all involved files to the project. This file has a text form and includes a specification of source sheet (BCS), tools (TOL), process behavior (PCT), project file (TFF) and other related files describing input parameters needed for the solver of

simulator. The next important files are defined with the numbered extensions 000 - 999 and they contain data in a binary format for all simulation steps.

The SIMViewer application allows taking advantage of advanced graphical and analyzing features to get required details of the simulation case study. For example, there are some disposable features intended for handling of the model on a scene, selectable cutting of the geometry parts, projecting of the sheet and each tool as a solid view, wireframe or combined both views. Furthermore, the application is able to provide a numeral and graphical analysis of wall thickness distribution, thermoforming temperature, stress, strain and other scalars in the place of defined cutting curve (see Fig. 3) that lies on the sheet in any selected step of the simulation process.

There are many other features in this application like an optional setting of the visibility, transparency, scalar scale ranges, background colors and color of each geometrical part on the screen. A user is also able to get details about particular nodes, elements or to obtain an overview about the currently displayed simulation step and project specification. This paper includes two video sequences to present particular simulation steps that unambiguously prove a gradual decrease of wall thickness in certain zones.

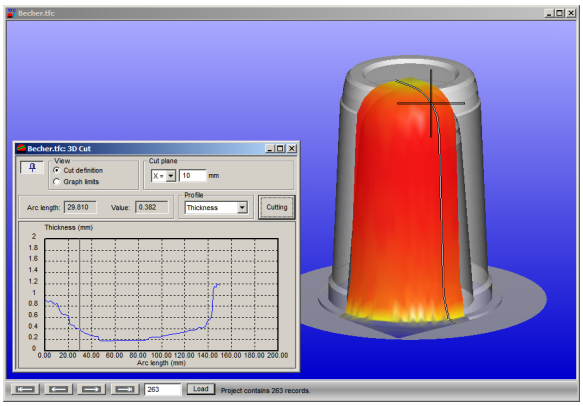


Figure 3. The analysis of deformed sheet by the cutting curve tool with the graphic chart.

5. SOFTWARE DEPLOYMENT

The SIMViewer application has been developed and deployed for interested students or customers who dispose of final data results produced by the T-SIM simulation software intended for thermoforming. The T-SIM simulator is fully commercial software with paid and registered license. If any user does not have this complex software installed on a computer by reason of that paid license, then he can simply use up this visualisation application. It is alternative software that is free of charge for anyone.

6. CONCLUSION

This contribution deals with the visualisation related to the thermoforming process simulation of plastic materials. The SIMViewer application was designed and built for the graduation theses at our faculty realized last year. This application processes specific data records incoming from the T-SIM simulation software developed by Accuform.

Technologists and people involved in the field of thermoforming work, they obtained an advanced visualisation application that is able to provide them various features for displaying, manipulating or editing and analyzing file data results from the solved project. The visualised data records represent an actual wall thickness distribution, thermoforming temperature, stress, strain and other scalars.

We have the next vision for an improvement of the application in the near future. It especially concerns on loading and displaying of 20 forming optional tools newly on the scene instead of 10 existing tools which represent an older data format. There also could be created a bounding box. This box should include the dynamic ruler in each axial direction to facilitate the exact measurement of sheet deforming through simulation steps (see Fig. 4).

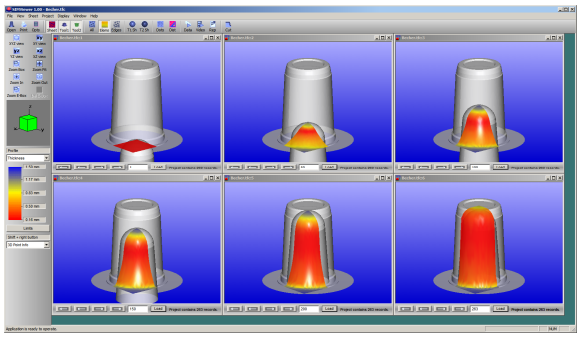


Figure 4. The thermoforming steps indicating their wall thickness distribution.

7. ACKNOWLEDGEMENTS

This work was supported by Accuform and Tomas Bata University, Faculty of Applied Informatics.

8. REFERENCES

[1] Thermoforming process. Sinotech, Portland, Oregon, United States, n.d. Web. 5 Mar 2013. http://www.sinotech.com/thermoforming.html

[2] Novotný, Petr. Large deformation of viscoelastic materials. Ph.D. thesis. Tomas Bata University in Zlín, 2004. Print.

[3] Gerža, Michal. Mathematical Modelling Results Visualization of Glass Furnaces. MA thesis. Tomas Bata University in Zlín, 2010. Print.

STRUCTURAL ENGINEERING IN TERMS OF DENSED HOUSING SYSTEM WITH ALLOWANCE FOR INSOLATION

Alexander L. Kheyfets
South Ural State University, Chelyabinsk, Russia
heifets@yandex.ru

ABSTRACT

The problem of a new building system within the existing residential area is considered in the article. Models and algorithms that help to calculate allowable space for a new building of a free form that won't reduce insolation of the buildings below allowable standards, are developed. Algorithm which helps to provide necessary insolation of a new building is considered. This model is realized by means of AutoCAD with AutoLisp. Algorithm for a model as well as the example for the calculation is given in the article.

Keywords

insolation, structural architectural design, computer simulation, geometric simulation, 3D-technologies, AutoCAD, AutoLisp.

1. INTRODUCTION

The erection of new buildings in the areas of constructed residential areas increases the role of the calculation of duration of insolation as a geometric factor that constraints the housing density. Modern methods and algorithms [1–3] allow us to calculate insolation by means of successive approximation only which makes this calculation quite problematic.

The objective of this article is to develop an algorithm of insolation calculation in terms of densed housing system.

Let us consider two problems. *Problem 1:* determination of allowable space for housing system, in which the erection of a building of any form will not reduce insolation of old buildings that are below allowable standards [1]. *Problem 2*: provision of allowable insolation of a new building under the influence of old buildings.

Methods of solution are based on the analysis and use of 3d-computer geometric simulation by means of AutoCAD [4]. The calculation of length and area within insolation boundaries is performed with the program [5].This algorithm is implemented by AutoLisp. This model is considered on the basis of typical housing development.

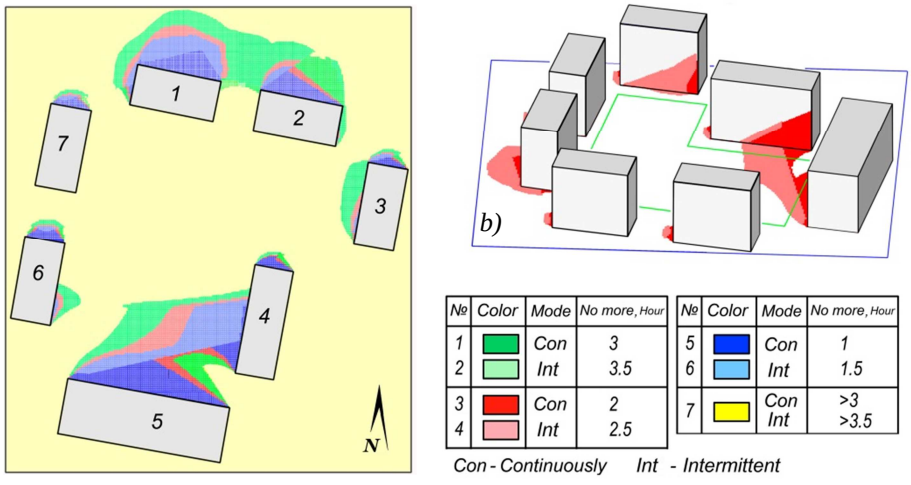
2. ALLOWABLE AREA FOR HOUSING DEVELOPMENT

The calculation of insolation boundaries of residential area (fig. 1) demonstrates inner zones of high insolation. This fact determines the possibility of placing a new building within this zone.

We construct the frame k for housing development and the overall volume V by vertical pulling of k (fig. 2, *a*). We place check points of the windows (fig. 2, *b*), in which we need to keep allowable insolation.

The calculation is performed on the day of equinox. The flux of sun rays directed in each check point presents a radial plane γ (see fig. 3, *a*). Angle φ is equal to geographic latitude of the area. In each point, let us take point A as an example, we build section of the buildings by radial plane and define sector L of insolation acting till the erection of a new building. Single sector β is formed for continuous insolation (fig. 3, *b*). Interrupted insolation with several sectors L is possible.

Within L we shall mark allowable insolation D, D_1, D_2, (fig. 3, *c*, *d*). Ray volumes V_d (V_1, V_2 …) are formed by the vertical pulling of D sectors. After subtraction of V_d from V we have volume V', provid-



a) *c)*

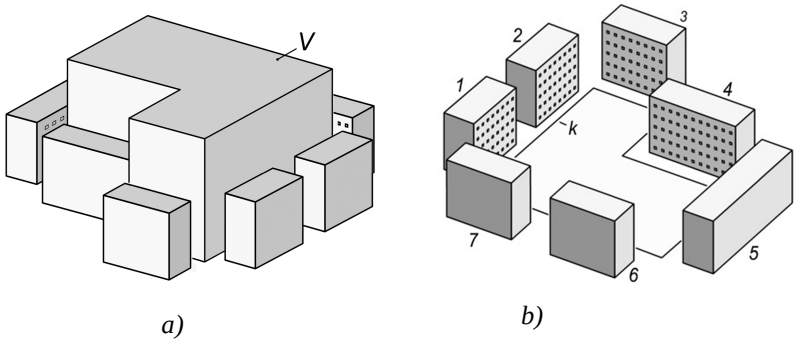
Fig. 1. Model of a residential area and its initial solar exposure:
a – field of solar exposure on the scheme; *b* – solar exposure of the walls (zones №. 3, 4); *c* – shading of the zones of solar exposure

 ISBN 978-80-86943-76-3

ing insolation in point A (fig. 3, *e*, *f*).

Within sector L we can assign either single sector D, lasting for 2 hours (30°) and creating a continuous residual insolation (see fig. 3, *c*), or several sectors D with a total duration of 2.5 hours (37,5°), creating disruptive insolation (see fig. 3, *d*). Apart from the quantity we can vary the placement of D inside L.

Within the frame of housing system k we define the set of basic points P_{bi} on a uniform grid (some points P_{bi} are given on



a) *b)*

Fig. 2. **Initial volume of a housing system** (*a*), **check points of the windows** (*b*)

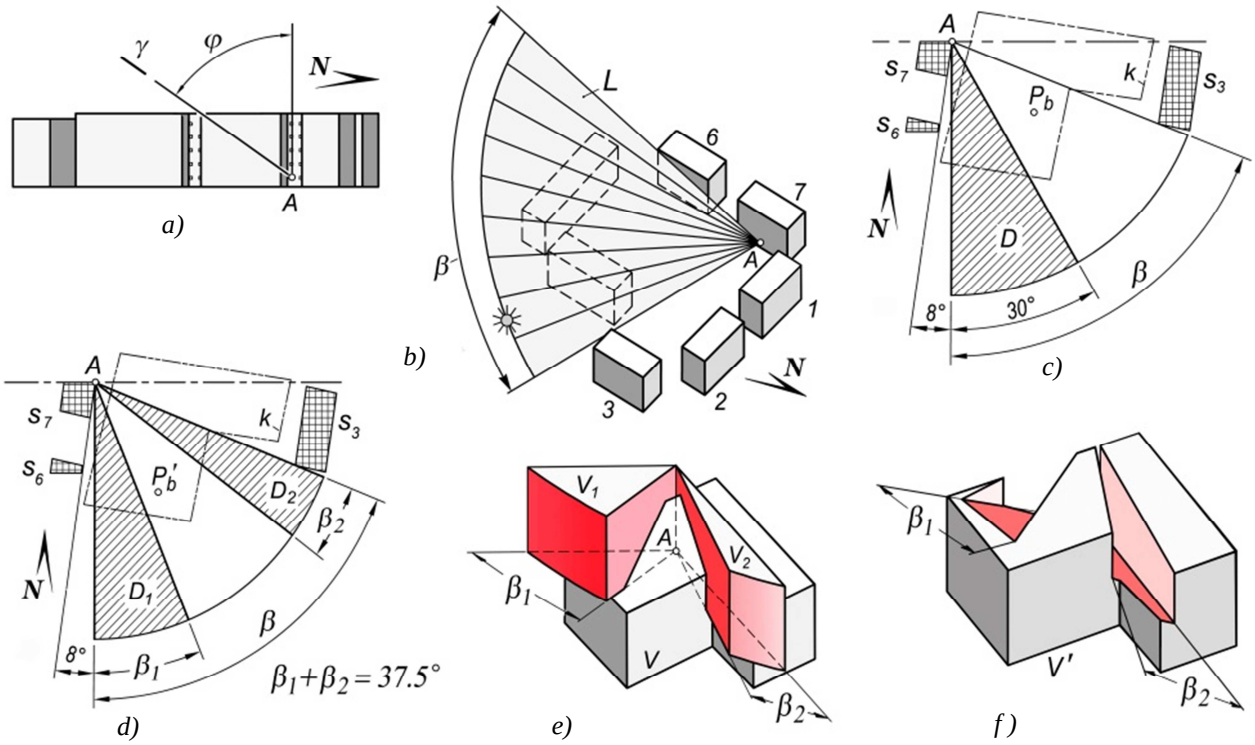


Fig. 3. **Algorithm of Problem 1:**
a – planes of equinox; *b*, *c*, *d* – sectors of permitted solar exposure at point *A*; e, *f* – subtraction of volumes of permitted solar exposure

fig. 4, *a*). For each point P_{bi} we form the set of sectors D of all check points, providing the area R_i of a maximum free space around P_{bi} .

Among the range of basic points we shall find point P_{bmax}, its area R_{max}, is a maximum one (fig. 4, *b*). Point P_{bmax} is considered to be optimal. Ray volumes V_d for all check points, defined for point P_{bmax}, are subtracted from initial volume V of housing development.

Final volume is a multifaceted body (fig. 5, *a*). After cutting off an unconstructive part the design of a new building is done (fig. 5, *b*), which guarantees allowable insolation at check points of old buildings.

In the given example the number of check points (windows) is 250, while the number of basic points on the frame is 120.

2. INSOLATION OF A NEW BUILDING

Let us put the markers of check points on the walls – windows (in our example there are 400 markers, fig. 5, *b*). At each point we replace the wall by the screen. We shall calculate insolation at the point. If it is of allowable level, we shall keep the position of the point without any changes. If insolation is below allowable level, we move the point, for example C (fig. 6, *a*), located on the wall w, into the depth of a building together with its screen e. Trajectory of movement m is perpendicular to the plane of the wall. The point is moved until insolation reaches its allowable level. We shall mark sector of allowable insolation D_c, and form volume V_c, , let us subtract it from the volume of initial model (fig. 6, *c*). For point C^* volume V_c (see fig. 6, *c* volume V_c is relatively raised up) is obtained by pulling frame D_c, down at value d, it equals the hight of the floor, and by pulling it up above the level of

 ISBN 978-80-86943-76-3

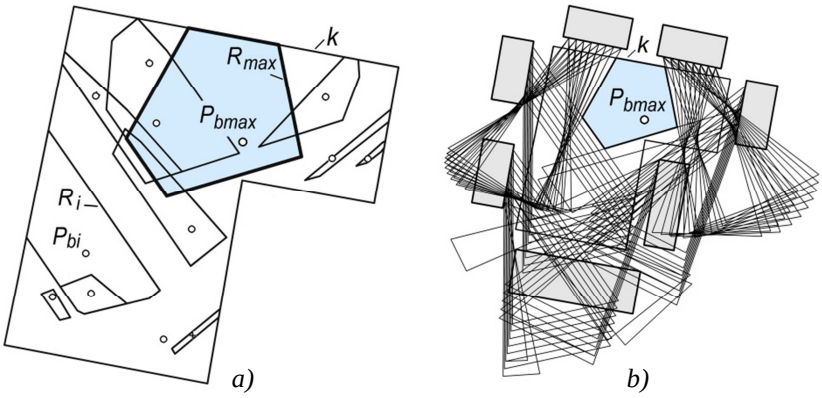


Fig. 4. **Algorithm of optimization of Problem 1:**
a – high-bay areas; *b* – sectors of light for optimal basic point

building area it will also provide its complete insolation. In the office area we provide parking space and warehouse where standard insolation is not required.

Fields of insolation demonstrate (fig. 7, *b*), that insolation of walls of old and new buildings is in allowable level.

3 ALGORITHM OF PROBLEM 1

Calculation of allowable volume of the housing development system is the problem of optimization. Its analytical solution is not possible due to nonlinear effect of the range of parameters.

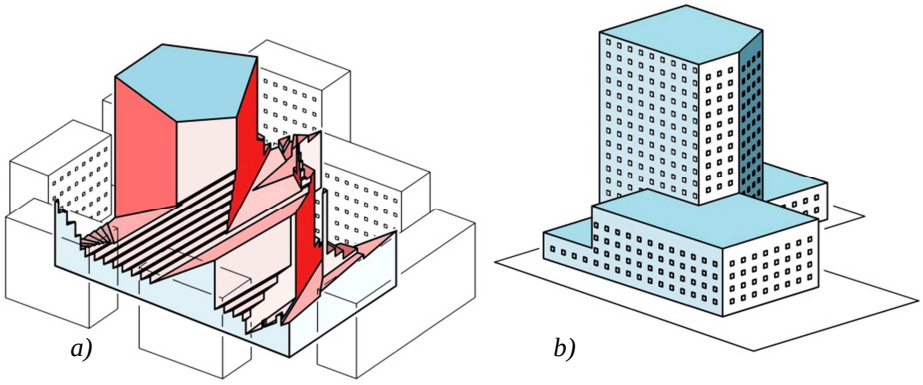


Fig. 5. Problem 1: *a* – overall volume; *b*– model inserted in volume

Algorithm which is connected with the selection of parameters of sectors D at every check point of the window P_k and calculation of optimal result in terms of placing a new building with a maximum allowable area of plan S for a high area is developed. Parameter of optimization is basic point P_b which is moved within frame k (fig. 4, *b*).

G is basic data to calculate geometry of old buildings, k is frame, V is initial volume, $MPk(m)$ is a massive of check points - windows), $MPb(n)$ is a massive of basic points.

At every point of a massive $MPk(m)$ we calculate sector L of initial insolation (see. fig. 3, *b*) and form a massive $ML(m)$ of the parameters for the sectors.

the model. After subtracting all the volumes of V_c we shall have a model. (fig. 6, *d*).

According to Problem 2 some changes of the model and organizational actions are necessary. Some faces of the office and high part of a building are turned at 20…30° (fig. 7, *a*). Although it will lead to the reduction of

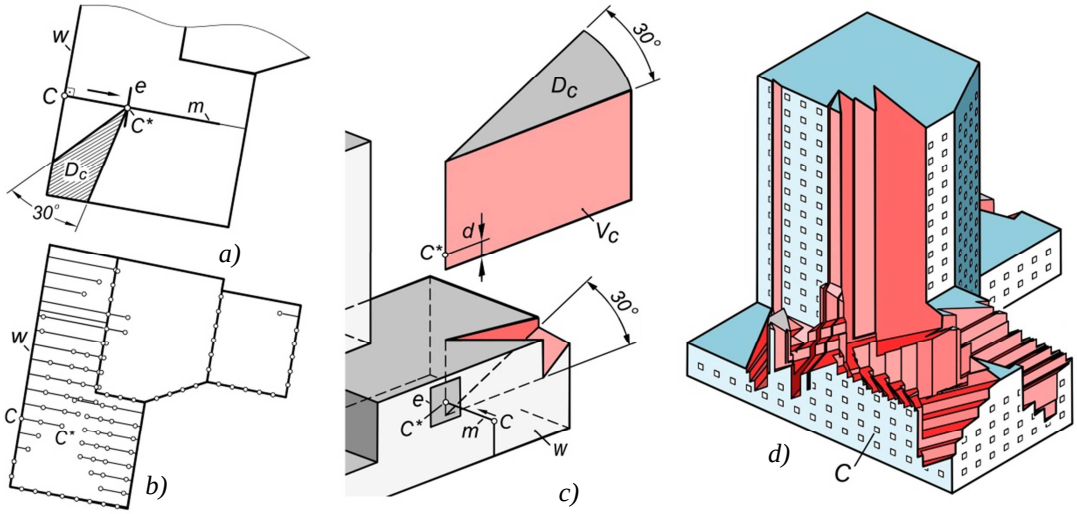
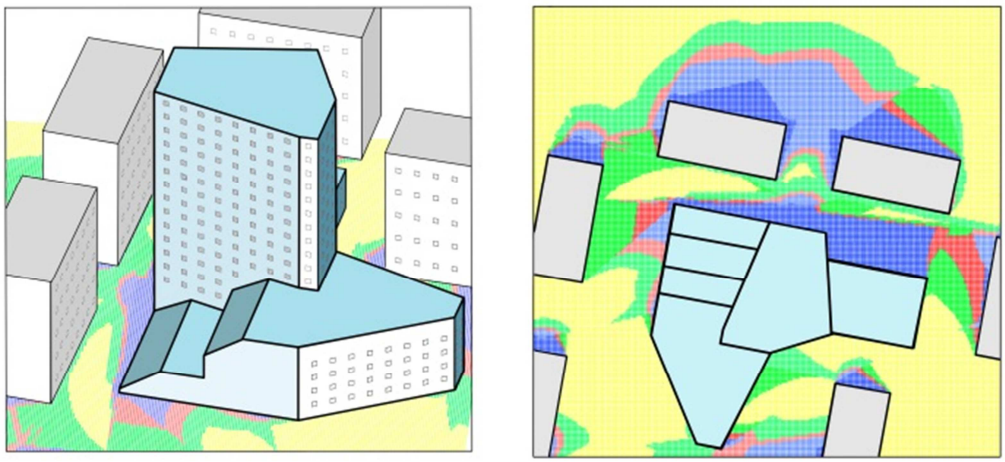


Fig. 6. Algorithm of Problem 2: *a* – formation of sector of permitted solar exposure; *b* – movement of check points; *c* – subtraction of sector at point *C; d* – subtraction of all sectors

 ISBN 978-80-86943-76-3



a) *b)*

Fig. 7. Final adjustment of the model of a new building:
a – model; *b* – solar exposure after adjustment

For every basic point P_{bi} of a massive $MPb(n)$ we shall form a massive $MD(m,i)$ of sectors D of all points of a massive $MPk(m)$. We shall calculate sectors D to provide area R_i of a maximum allowable area S_i round the points P_{bi} (see fig. 4, *a*).

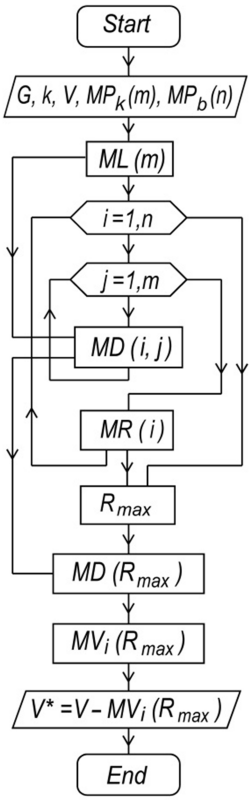


Fig. 8. Block scheme of problem 1

Algorithm for sectors D formation provides 20 variants of their placement within sectors L depending on the size of the sectors L, type of initial insolation (*Con*, *Int*, see fig. 1) and the position of basic point P.

Let us insert parameters $MD(m,i)$ to massive $MD(m,n)$. We shall insert geometry of areas R_i (see fig. 4, *a*) of all basic points to massive $MR(n)$.

In a massive $MR(n)$ we shall find area R_{max} of a maximum area S_{max} and corresponding basic point P_{bmax}, which are considered to be optimal. From massive $MD(m,n)$ we shall extract $MD(Rmax)$ of sectors D to the area R_{max}, we shall build ray volumes from these sectors, and form a massive $MV(Rmax)$, which is to be subtracted from basic volume V. Volume V^* obtained is considered to be the solution of the problem 1.

SUMMARY

1. Model under consideration and algorithms of 3d-computer geometric simulation for its implementation allow defining allowable volume of housing system under guaranteed insolation of new and old buildings.

2. Fields of insolation are an effective and strategic geometric method of its analysis.

3. The model given shows the possibilities of modern methods of 3d computer geometric simulation in the solution and analysis of the problems, for which the use of traditional analytic or project methods is impossible or irrational.

REFERENCES

[1] Hygienic requirements for insolation and solar control premises of inhabited and public buildings and grounds [Sanitarnye pravila i normy SanPiN 2.2.1/2.1.1.1076-01: Gigienicheskie tre-bovanija k insoljacii i solncezashhite pome-shhenij zhilyh i obshhestvennyh zdanij i territorij: utv. Glavnym gosudarstvennym sanitarnym vrachom RF 19 oktjabrja 2001 g.]: http://best-stroy.ru/gost/r38/311.

[2] Sitis: Soljaris 5.15.11311 (17.08.2011): http://sitis.ru/soft/solaris.

[3] Kheyfets A.L. Insolation Duration Calculating by Means of 3D-Modelling Program Autocad. 7th International Conference 3IA'2003. 12-13 of May. 2004. Limoges. France. Computer Graphics and Artificial Intelligence. Organised by MSI Laboratory of the University de Limoges. pp. 177–180.

[4] Kheyfets A.L, Loginovskij A.N., Butorina I.V., Vasil'eva V.N. Engineering 3d computer graphics [Inzhenernaja 3D-komp'juternaja grafika. AutoCAD]. Moscow, Izdatel'stvo Jurajt, 2011. 464 p.

[5] Kheyfets A.L. The program of the automatic calculation of the insolation [Programma avtomaticheskogo rascheta insoljacii]. Svidetel'stvo o gosudarstvennoj registracii programmy JeVM №2010613828. Zajavka № 2010612286. Data postuplenija 23 aprelja 2010 g.

Compressed Grids for GPU Ray Tracing of Large Models

Vasco Costa
INESC-ID/IST
Rua Alves Redol, 9
1000-029 Lisboa,
Portugal
vasco.costa@ist.utl.pt

João M. Pereira
INESC-ID/IST
Rua Alves Redol, 9
1000-029 Lisboa,
Portugal
jap@inesc-id.pt

Joaquim A. Jorge
INESC-ID/IST
Rua Alves Redol, 9
1000-029 Lisboa,
Portugal
jaj@inesc-id.pt

ABSTRACT

Ray tracing on GPUs is an area of ongoing research. GPUs are well suited for this parallel rendering algorithm. GPU computing devices typically have characteristics which make them quite different from CPUs: increased data parallelism, increased memory bandwidth, smaller caches, lower memory capacity. Presently it is difficult to visualize large scenes with tens of millions of triangles in these memory constrained platforms. In this paper we present a compressed grid data structure, capable of state of the art rendering performance, using up to 6× less memory than conventional grid storage schemes. The compressed grid is built and traversed on the GPU.

Keywords

Ray-tracing, gpu, grid, compression.

1 INTRODUCTION

Display devices have been increasing in resolution at a more rapid pace than in the past. This means scenes with low polygon counts are no longer suitable as users can perceive the large polygons therein thus reducing their level of immersion. In addition the ray tracing rendering algorithm has been gathering increased attention. It is possible to ray trace complex scenes at real-time frame rates on a single GPU of the latest generation.

The ray tracing algorithm is more amenable for photorealism as it is a global illumination algorithm which can easily display shadows, reflections, or refractions. It is possible to extend it for diffuse interreflections as well using path tracing, or photon mapping albeit at much reduced frame rates. In this paper we focus on solving the basic ray shooting algorithm which is used for all these cases.

To provide real-time ray tracing performance an acceleration structure must be employed in order to reduce the number of ray/polygon intersection tests required to render the scene.

Existing work for GPU ray tracing includes [LGS+09, PL10] which focuses on bounding volume hierarchy (BVH) acceleration structures, [HSHH07] which describes kd-tree acceleration structures, and [KBS11] which focuses on grid acceleration structures. Other interesting developments on CPU ray tracing include: [Áfr12] which implicitly stores a BVH acceleration structure with reduced space requirements (still it uses temporary storage for bounding boxes and other auxiliary data), [LD08] which employs row displacement compression to reduce grid memory storage requirements.

Figure 1: The Lucy model (28 Mtri) rendered at 1024×1024 resolution. The grid acceleration structure is compressed to 210.45 MB. An uncompressed uniform grid of the same dimensions uses 1258.48 MB of memory.

 ISBN 978-80-86943-76-3

Our work implements row displacement compressed grids in streaming computing architectures such as GPUs.

The organization of this paper is as follows: we describe a novel parallel algorithm for construction of row displacement compressed grids on the GPU, next we analyze the algorithm's performance compared to state of the art grid implementations.

2 GRIDS

Uniform grids are spatial partitioning structures which divide space into identical cubically shaped cells also named voxels in the literature. Typically a grid construction algorithm first computes the scene bounding box then uses an heuristic to compute the number of grid split planes along each major axis of the scene bounding box. These heuristics commonly attempt to use an amount of space directly proportional to the number of primitives in the scene. Thus we arrive at the following heuristic common in grid literature:

$$M_i = S_i \sqrt[3]{\frac{\rho N}{V}} \quad i \in \{x, y, z\}$$

Where ρ is the grid density parameter which in our case is equal to 5. The number of cells M is equal to the grid resolution $M_x \times M_y \times M_z$. N is the number of objects in the scene. S_i is the scene bounding box size in dimension i. V is the bounding box volume.

Ray shooting is implemented by traversing the grid cells intersected by a ray [AW+87] from its point of entrance to its point of exit.

For typical scenes most uniform grid cells will be empty. This means some form of sparse matrix compression scheme is desirable. In our case we implemented the row displacement compression scheme, represented in Figure 2, described in [LD08].

The process of computing the offsets for each row can be parallelized in the GPU as can the other steps of uniform grid construction. Hence we arrive at Algorithm 1.

3 METHOD

The row displacement compression method stores grid rows, in an overlapped fashion, inside a 1D array L. A 2D array O stores the offsets to the start of each grid row inside L. Prior to accessing this hashed grid a 3D

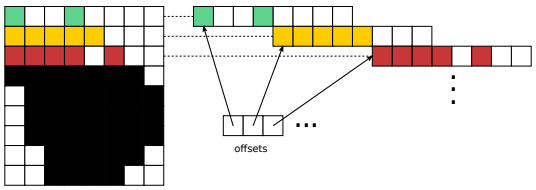


Figure 2: Row displacement compression.

Algorithm 1 Parallel compressed grid build.

```
1:  function BUILDCOMPRESSEDGRID(M, objects)
2:      D ← DOMAINBITS(M, objects)
3:                  ▷ population count followed by reduce
4:      nonEmpty ← NONEMPTYCELLS(D)
5:      NO, NbH ← M_y × M_z, 2 × nonEmpty
6:      O, last ← FILLOFFSETS(NO, NbH, M, D)
7:      NH ← last + 1
8:      H ← COMPUTEOFFSETS(NH, M, O, objects)
9:                                  ▷ inclusive scan
10:     NL ← COMPUTEPREFIXSUM(H)
11:     L ← INSERTINDICES(NL, M, O, H, objects)
12:     return D, O, H, L
13: end function
```

bit array D, also known as the domain bits array, is consulted to determine if that particular cell is occupied.

In our implementation the domain bits, which state if a grid cell is empty or not, are stored as a linear bit array. Internally the bit array is composed of unsigned ints with 32 bits each. Domain bit computation, as other steps in the algorithm, is made in parallel: for all objects in the scene we determine which cells they overlap, then insert then into the domain bits with atomic memory operations. The computation of the number of non-empty cells is done by a population count pass, followed by a scan pass.

Row displacement compression offsets are computed in the next step and stored in the O array. This step of the algorithm is computationally expensive since it computes a mapping of the grid rows into a compressed 1D array using a find-first-fit scheme. The H array is computed by storing the number of objects which overlap each cell with atomic memory operations. The prefix sum is then computed so each cell points to the tail of its item list. Finally object indices are inserted into the item list with atomic operations. The atomic locks have a fine granularity in order not to constrain parallelism.

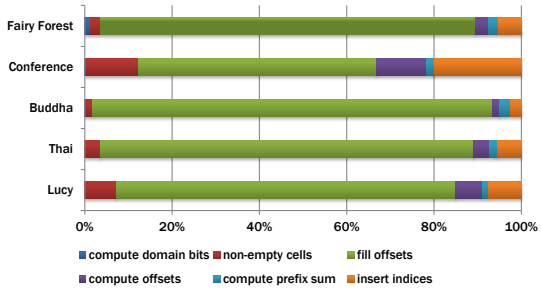
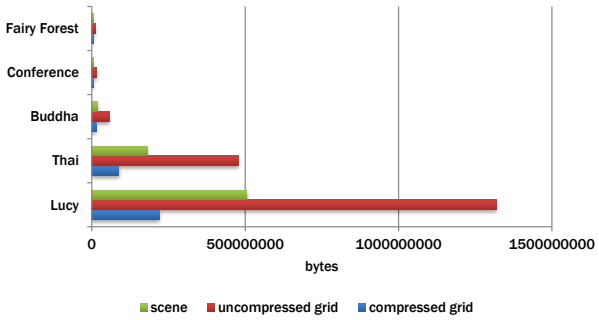


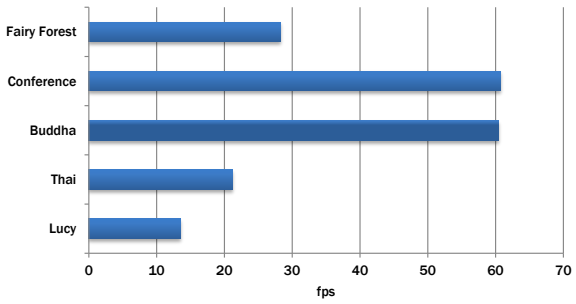
Figure 3: Percentage of time spent in each step of GPU grid construction.

As can be seen in Figure 3 the most time consuming operation is the FILLOFFSETS step where row offsets are computed using the first-fit method.

 ISBN 978-80-86943-76-3



(a) Memory used to store the scene and the grid acceleration structure with and without compression.



(b) Framerates for selected scenes.

Figure 4: Memory consumption and rendering frame rates for the test scenes.

The singled-threaded CPU implementation of this algorithm has better performance for scenes with small grid row M_x sizes since these are mostly serial workloads where it is not possible to extract enough row parallelism for the GPU to prevail. However as can be seen in Figure 5 the GPU implementation dominates for the larger scenes with more geometry and correspondingly larger row sizes.

4 RESULTS

Our test platform is an AMD FX 8350 8-core CPU @ 4.0 GHz powered machine with 8 GB of RAM. The graphics card includes a NVIDIA GeForce GTX 660 Ti GPU with 2 GB of RAM.

The implementation language is ANSI C++ for the host code and OpenCL running on the GPU for the compute kernels. During rendering the local work group size is set to 16x16 blocks in order to maximize cache locality and take advantage of pixel parallelism. The application runs on the Linux operating system. All images were rendered at 1024×1024 resolution using one ray per pixel and diffuse shading.

Ray/triangle intersection is done with the Möller-Trumbore algorithm [MT97] since it does not require the usage of any additional memory. Each triangle uses 12 bytes of memory to store the vertex indices and each vertex uses 12 bytes of memory. For scenes with normals each vertex normal also uses 12 bytes of memory.

The Fairy Forest and Conference scenes are representative of the scenes you can typically find in a computer game with irregular polygon density i.e. high polygon count objects inside a lower polygon count environment with walls. The Buddha, Thai Statue, and Lucy models represent scanned scenes with triangles of similar area. These scanned scenes feature larger total polygon counts than the first two.

As can be seen in Table 1 our algorithm has good rendering performance compared to previous work on GPU single-level grids [KS09] and two-level grids [KBS11]. This is probably due to our algorithm having improved cache coherence. Our algorithm requires less memory bandwidth per cell traversal. The GPU we are using, the GTX 660 Ti, has similar bandwidth compared to the earlier GTX 480. However the GTX 660 Ti has much improved peak floating point performance making it hard to judge the improvement of the work based on the strengths of a software implementation alone. Two-level grids typically have better render time performance than single-level grids. However our single-level grid implementation on a GTX 660 Ti has better rendering performance than the previously

Scene	Grid GTX 280	2lvl Grid GTX 470	Compressed Grid GTX 660 Ti CPU	GPU
Fairy Forest	24 ms 3.5 fps	8 ms 21 fps	20 ms 28 fps	65 ms
Conference	27 ms 7.0 fps	17 ms 26 fps	48 ms 61 fps	78 ms
Thai Statue	417 ms -	257 ms -	537 ms 21 fps	375 ms

Table 1: Performance comparison of our Compressed Grid implementation, with the Grid from [KS09], and the 2lvl Grid from [KBS11]. The table lists grid build times and rendering frame rates. The Thai Statue scene frame rate performance was not specified in those articles. For the Compressed Grid the CPU and GPU implementations of FillOffsets were tested on grid construction.

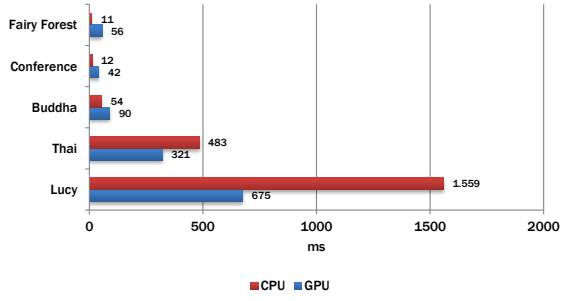


Figure 5: Time required to fill the offset table using the GPU vs the CPU.

 ISBN 978-80-86943-76-3

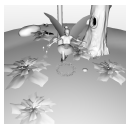
	 Fairy Forest	 Conference	 Buddha	 Thai Statue	 Lucy
Scene					
Triangles	173.98 K	282.76 K	1.09 M	10.00 M	28.06 M
Memory	4.21 MB	5.15 MB	18.67 MB	171.66 MB	481.60 MB
Uncompressed Grid					
Memory	11.22 MB	15.44 MB	53.50 MB	455.90 MB	1.22 GB
Compressed Grid					
Memory	5.41 MB	5.44 MB	13.84 MB	83.85 MB	210.45 MB
Build Time	64.77 ms	77.83 ms	98.68 ms	375.08 ms	870.15 ms
Frame Rate	28.30 fps	60.72 fps	60.40 fps	21.22 fps	13.52 fps

Table 2: Scene statistics and grid performance results versus an uncompressed grid data structure [KS09].

mentioned two-level grid implementation on a GTX 480.

The advantages of the compressed grid are more debatable on grid construction. While our implementation features better performance than previous work on larger scenes with tens of millions of triangles, like the Thai Statue, it has poorer performance on the lower polygon scenes. This is due to the time spent performing grid compression namely the FillOffsets phase, as seen in Figure 3, where rows offsets are computed using a first-fit method. For such low polygon count scenes a single-threaded CPU implementation has better performance than our GPU kernel as can be seen in Figure 5. While build times are not an issue for static scenes this remains an open problem in this GPU grid implementation.

Our most important objective is the reduction of memory consumption in order to enable the visualization of larger scenes. As can be seen in Table 2 the implemented single-level compressed grids feature much lower memory consumption than uncompressed single-level grids. Thus compressed grids enable the visualization of more complex scenes. In our case we can visualize the Lucy statue using a sixth of the memory required for a grid without compression as can be seen in Figure 4a.

5 CONCLUSIONS AND FUTURE WORK

This work enables the visualization of large scenes, with tens of millions of triangles, on the GPU. The acceleration structure construction and rendering is performed in parallel in the GPU. The algorithm provides real-time frame rates for scenes with millions of triangles.

For dynamic scenes with destructible geometry we require more rapid grid construction times. This may be achieved with alternative hashing or sparse matrix storage algorithms. This work also does not have optimizations for highly coherent rays such as the use of ray bundles.

6 ACKNOWLEDGEMENTS

This work was supported by national funds through FCT - Fundação para a Ciência e Tecnologia, under project PEst-OE/EEI/LA0021/2013.

We would like to thank the Stanford 3D Scanning Repository (Buddha, Thai Statue, Lucy), the Utah 3D Animation Repository (Fairy Forest), Anat Grynberg and Greg Ward (Conference) for the test scenes.

REFERENCES

[Áfr12] A.T. Áfra. Incoherent ray tracing without acceleration structures. In *Eurographics - Short Papers*, pages 97–100. Eurographics Association, 2012.

[AW+87] J. Amanatides, A. Woo, et al. A Fast Voxel Traversal Algorithm for Ray Tracing. In *Proceedings of Eurographics*, volume 87, pages 3–10. Eurographics Association, 1987.

[HSHH07] D. R. Horn, J. Sugerman, M. Houston, and P. Hanrahan. Interactive k-D Tree GPU Raytracing. In *Proceedings of the 2007 Symposium on Interactive 3D Graphics and Games*, pages 167–174. ACM, 2007.

[KBS11] J. Kalojanov, M. Billeter, and P. Slusallek. Two-Level Grids for Ray Tracing on GPUs. In *Computer Graphics Forum*, pages 307–314. Wiley Online Library, 2011.

[KS09] J. Kalojanov and P. Slusallek. A Parallel Algorithm for Construction of Uniform Grids. In *Proceedings of the Conference on High Performance Graphics*, pages 23–28. ACM, 2009.

[LD08] A. Lagae and P. Dutré. Compact, Fast and Robust Grids for Ray Tracing. In *Computer Graphics Forum*, pages 1235–1244. Wiley Online Library, 2008.

[LGS+09] C. Lauterbach, M. Garland, S. Sengupta, D. Luebke, and D. Manocha. Fast BVH construction on GPUs. In *Computer Graphics Forum*, pages 375–384. Wiley Online Library, 2009.

[MT97] T. Möller and B. Trumbore. Fast, Minimum Storage Ray-Triangle Intersection. *Journal of Graphics Tools*, 2(1):21–28, 1997.

[PL10] J. Pantaleoni and D. Luebke. HLBVH: Hierarchical LBVH Construction for Real-Time Ray Tracing of Dynamic Geometry. In *Proceedings of the Conference on High Performance Graphics*, pages 87–95. Eurographics Association, 2010.

Architectural Library. Dioscuri Temple in Agrigento

Laura Inzerillo
DARCH Palermo University
Viale delle Scienze ed. 8
90100, Palermo, Italy
laura.inzerillo@unipa.it

Giuseppe Dalli Cardillo
PhD Palermo University
Viale delle Scienze ed. 8
90100, Palermo, Italy
dacar@libero.it

ABSTRACT

This paper presents the synthesis of research about the ICT solution for cultural heritage through the innovative technology of representation and survey. One objective was to identify a method of processing data for knowledge, dissemination and preservation of the asset. Historical analysis, surveying, photogrammetry and laser scanner, the processing of survey data and modelling are all focused to the same final purpose. The methodology developed for this application framework becomes adoptable in General for Cultural Property, which become in turn the protagonists of 3D WebGIS.

Keywords

Architecture, 3D web, Geometry, 3D modelling, Dissemination, Communication, AR.

1. INTRODUCTION

In recent years, the problem of communication for the conservation of cultural heritage has been tackled.
The literature today requires the use of the web as a primary tool for the *dissemination*, *communication*, storage and knowledge on architectural buildings.
Through the case study examined, the Temple of *Castor* and *Pollux* in the Valley of the Temples in Agrigento, the *technological know-how* of communication, required to have the adeguate cultural diffusion of a building of architectural or archaeological value, has been determined in terms of *accessibility*, *queryability*, *manageability* and *interactivity*. 3D modeling is an essential element in the understanding, enjoyment, and historical analysis of both architectural and archaeological cultural heritage. The creation of a 3D model involves applying a methodology that, inevitably, assumes that the operator that makes the model is a masters in history, geometry, in metric, compositional, architectural, archaeological, and semantics of the cultural architectural heritage. However, it is not generally the case that the operator coincides with the researcher or with whomever has the critical skills for the realization of a functional and usable model. The steps leading to the creation of a 3D model are well known to researchers in the field of representation, and are easily understood even for non-experts. In fact, the operator must know the cultural architectonic heritage both historically and then metrically, to be able to represent it. To understand well and know a cultural heritage metrically it is necessary to apply survey technologies such as the topographic method, the normal or stereo photogrammetric method for texture mapping or laser scanner for point clouds. Mastery of the technology allows the user to create effective items within a reasonable time. Skills in the field of survey must be accompanied, unavoidably, by an upgrade of the data collected. Once the model is developed, geometrically correct, metrically accurate, and aesthetically perfect, one wonders what the main aim is; what purpose does this small object serve? It is difficult to manage due to its size in terms of bytes, and it is so beautiful that you are almost afraid to manage it. So we have arrived at the focal point of the matter: the purpose. Yes, we could talk about goals aimed at historical knowledge of cultural property; or goals aimed at the disclosure of architectural geometry; or, again, of the strata, of anastylosis, of typological-semantic compositions, of structural conditions, of wall cladding, of artistic value, of restoration... Thus the model is made to answer one or more of such questions, and the user would be the recipient and unique depositary of the asset. But this cannot and must not satisfy the thirst for knowledge, and especially it must not silence the conscience that now, more than ever, needs to provide a future for future generations, the same vision that for us has been a source of cultural, ethical, aesthetic, poetic and ascetic inspiration. It means that, with a view to sustainable development,

cultural heritage is not an element that is separate from the context, is not an end in itself, but belongs to a network of connections, a woven fabric of relationships, an economic system. This cultural heritage belongs to the community, but only for a limited and short-sighted period until the asset loses some of its value and is relegated to the few fans who still find it a value. Architectural heritage belongs to the historical memory of each of us and, at the same time, the future memory of those who can still enjoy it and make it part of an accomplished life. We must, therefore, enhance architectural heritage through extended and wide-spread dissemination that is for everyone, experts and non-experts. Architectural heritage must enter people's homes to be appreciated by presenting graphical models that are *easy* to *manage*, *easy* to *use*; where a click is the link that satisfies our thirst for *practical information* (on opening times, cost of tickets, guided tours, group discounts, etc.), *generic information* (history, iconography, structural layers and/or anastylosis through real-time animation, virtual tours within the structure, immersive paths in *AR*, typological architecture or archaeology, marble coverings, paintings, authors, etc.), or *specialized information* (geometric sections that fetch moving plans with a choice of static characteristics, examination of the foundations, monitoring the state of conservation of the property, mapping of the restoration, calculation of surfaces, etc.).

PILOT PROJECT

We identify a pilot project that became the point of departure with the past, in order to maintain and preserve the cultural heritage for future generations.

When we talk about the pilot project, of course we must consider an important transaction that involves the whole relevant scientific community, both nationally and internationally. We refer to the substantial problem of making the data consistent, *accessible*, *interpretable*, *usable*, and *processable* from all points of view and in all scientific fields. The homogeneity affects all sectors, not only thematic areas such as history, restoration, geometry, etc., but the *homogeneity* of *navigational queries*. In essence, it is the realization of a *framework* that involves itself in every aspect of the development of a model, a method. When constructing a pilot project we have to take account of *benchmarks* that overall make the product *universal* and *accessible*. Some of these parameters are easy to understand, for example think of a timeline or ranking. It is clear that a homogeneous product has a similar load time: it is unthinkable that there may be a very long load time to download sizeable models that are difficult to manage. It is also true that in a project that is intended as a global product, there is a set of webpages that follow similar philological paths according to a typological classification for the ease of identification. The purpose of the project should be recalled: it is necessary to return to the *diffusion* of the *knowledge* of the *cultural architectonical heritage* through its facilitated and simplified disclosure to the point that the cultural heritage itself becomes part of the everyday. This creates the preconditions to understanding the cultural heritage, the story of its past, and also of his future through conservation planning and effective communication.

Let us look more closely at the methodology required, starting from the architecture's history and ending with the creation of 3D WebGIS.

All material on all pages should fit within a rectangle of 16 x 23.7 cm (6.3"x 9.33"), centered on the page horizontally, beginning 2.5 cm (1") from the top of the page and ending with 3,5 cm (1.4") from the bottom. The right and left margins should be 2.5 cm (1"). The text should be in two 7.6 cm (3") columns with a 0.8 cm (0.3") gutter.

2. THE SURVEY

Historical and iconographic analysis

Agrigento was founded on a plateau overlooking the sea, with two nearby rivers, the Hypsas and the Akragas, and a ridge to the north offering a degree of natural fortification. Its establishment took place around 582–580 BC, and it is attributed to Greek colonists from Gela, who named it Akragas. Akragas grew rapidly, becoming one of the richest and most famous of the Greek colonies of Magna Graecia.

Topographic survey

Detecting architecture means rebuilding the concept of its designer, his technical construction choices and his realization strategies. The "measure" has fundamental value; technological advancement helps us achieve greater accuracy and speed. However, the measure must be supported by a pre-critical understanding of the cultural heritage; this knowledge can alone ensure the significance of the survey itself, a definition of high effectiveness and completeness that deserves careful analysis in order to understand the full meaning of its content. The survey is a set of operations of knowledge that contribute to the reconstruction of the life, history and lifecycle of an architectural building.

The procedure adopted, by applying the so called polygon technique, has allowed us to identify an appropriate number of consecutive fixed points (so that each point is visible from the following and the antecedent), therefore constituting the vertex of a closed broken whereby sides and corners are measured.

The photogrammetry and the photomodellation of built heritage

New technologies offer the ability to create and share knowledge. The digitization of heritage contributes to the development of the information world.
The current tools and techniques of relief and representation give access to results to date unimaginable about precision, speed, comprehensiveness and realism. We know the potential that photogrammetry offers to the survey. Today, photomodellation makes it possible to operate in a work space allowing the 3D restitution of the built heritage, based on the global integration between the phases of surveying, modeling and representation. Apply photomodellation to the 3D reconstruction of buildings implies that an operator must to employ an interpretative effort that involves an understanding of the architectural forms that make up the building.

Laser scanning and cloud points

Laser scanning of the temple of Castor and Pollux has produced a highly-detailed point cloud. The large number of information derivable from it is, however, proportional to its difficulty of management. In fact the high amount of data is both the point of force that criticality of the detection mode with laser scanner.
It was necessary to maintain a good approximation and precision with respect to the detected object without affecting the objective of research, i.e. to obtain a manageable model that is also interrogable in real-time. A often adopted solution is to decimate the point cloud, however in our case this was not advisable. In fact, the temple was built of sandstone, and if we had used the method mentioned we would have risked losing some important information.

3. OUTPUT FOR KNOWLEDGE

Post processing and testing

To overcome the difficulty of management we decided to use the technique of Retopology, well known in the cinema sector: which consists of a system that generates new polygons, a new mesh that adapts itself to the existing one which is the basis and the reference. The new polygons, which are generated manually with total control by the opera-tor, follow a new "topology", that is orderly, precise, and usually supported by a lower number of triangles: it is optimized according to the areas to "trace".
The creation of a mesh derived from a point cloud is often the result of complex and disordered triangulations. Retopology solves this problem through the reconfiguration of the geometric shapes that make up the cultural heritage through a rational division into triangles or quadrangles. In fact, the automatic algorithms are often designed in a way that even a very simple surface or plane surface will be described by an exorbitant number of polygons or triangles, making the portion in question unnecessarily rich in information while the same could be well described by a lower number of polygons without loss of quality.

Geometrical and 3D model

Such models are precise but also easy to manage, and allows a model to be developed for immersive and interactive visualisation systems, such as rendering and real-time augmented reality. These solutions have led to significant results, in both the scientific and disclosure aspects of the product, such as the ability to make the model of the temple "appear" superimposed on the real one, and allowing switching from one model to another at the touch of a button. Furthermore, you can also "walk" inside the building; and an experimental trial also included the possibility to make so-called *hotspots* appear – accessible via the mouse to some specific points of the model – providing additional retractable information, pop-ups, for example text analyses or pictures or video explanations. Now that we have a model corresponding to the requirements, some important elements to be added are *textured shaders* to improve the *real-time* visualisation of the cultural heritage.
In the field of optics, *shaders* are commonly used to simulate the diffusion, reflection, refraction and dispersion of light. The latest *Graphics Processing Units* has *papelines* used for transformation of a 3D scene in a raster image that can be represented on the screen. The *pipelines* carry out the operations based on programmable *shaders*.

3Dmodel for Google Earth

The model that is implemented in Google Earth is generally a model that responds to the generic requirements of modelling, including non-photorealistic aspects. A feature of the model is georeferencing, which responds to a link that refers to photographs and other links that lead to sites related to the cultural heritage. Each part was modelled correctly under the metric and geometric profile and trying to reconcile the complexity of the model with the limitations imposed by GE, for example the number of polygons of each three-dimensional element. The mathematical model is transformed into a numerical model. As the model further progresses, is also ensured that objects are represented in translucence, for example to highlight horizontal or vertical sections of the building's facades or of the model in anastylosis. The rendering and the creation of texture useful for interactive exploration were undertaken using Cinema 4D with Ambient

Occlusion. The georeferencing of the model and of each point on of the model's sides allows the querying of any part of the model with a click of the mouse. This work was developed in Andrea Palladio 3D Geodatabase, a geo-database of a three-dimensional web-based platform where 3D models integrate a large *Palladian Informative System*.

3D GisWeb

The study of a 3DGisWeb requires good knowledge and expertise to identify the appropriate communication channels in order to achieve the initial goals. Among the new generation of software you could include Geoweb3D, which combines the potential of the Web with the dell'ArcGisTabs and 3D. Progress in geospatial visualisation and analysis is all about making it easy to use, making it work with all available formats and maximising the effective communication of user data. The User Interface is extremely intuitive and is what app users now expect. Users do not need to know anything about 3D modelling. Load your data, style the appearance, save it and share it – that's it. Constructing those apps with endless small buttons, complex dialogues, and high learning curves is the opposite of what the user wants. Users no longer need to convert or restructure their data in any form. As the data changes, so will the visualisation. Most *raster*, *vector*, *kml*, *web services*, and *3D model* formats are supported. *ESRI* users can load *MXDs*, use their geodatabase and access *ArcGIS Server* content. Data is reprojected on the fly from any of the thousands of supported coordinate systems. *Geoweb3D* has been built from the ground up by leveraging the latest in video card *GPU technology* and developed exclusively for geospatial 3D visualisation and analysis; it is not a repurposed application nor CPU-reliant architecture developed before gaming technology was mainstream. This is the best of GIS and the best of 3D. Users need the seamless integration of the disparate GIS formats, feeds, containers and delivery mechanisms that vary among users, organisations and agencies. Geoweb3D, although supportive of standards and modern software design, will scale with the industry. Those users confident that a currently common format will not be replaced are those ones trying to find the DVD drive on a tablet.

The only constant in this industry is change.

4. CONCLUSIONS

The method applied, and outlined here, shows that the adopted system leads to space-time web collocation of the architectural heritage. The product is easy to access, enjoy, interact with, manage, monitor, thus ensuring disclosure and therefore safeguarding knowledge and conservation.

This means that the potential of the architectural heritage is valued to the extent that it is possible to fulfil the requirement of sustainability declared by the Bruntland Commission: "development that meets the needs of the present without compromising the ability of future generations to meet their own needs".

The architectural heritage belongs to a network of social, economic, architectural, archaeological, touristic and cultural movements. All these aspects are needed for the conservation of architectural heritage.

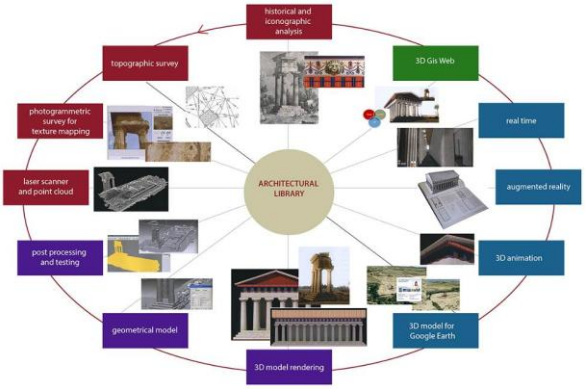


Figure 1. Know how: from history to the 3D WebGis.

5. REFERENCES

[Bel00a] Beltramini, G. and Gaiani, M. Dalla grammatica palladiana alla Palladio Library: piccola storia del sistema comunicativo-informativo palladiano, in Palladio Lab, Architetture palladiane indagate con tecnologie digitali, Vicenza, 2012.

[Inz01a] Inzerillo, Augmented reality: past, present, and future, in proceeding SPIE The Engineering Reality of Virtual Reality 2013, 86490E (March4, 2013), San Francisco, California, United States, 3-7 febbraio 2013, in PROCEEDINGS OF SPIE (ONLINE)] E215049 - issn: 1996-756X, doi:10.1117/12.2001833 - Vol. 8649, on line *Collana*. © (2013).

3-D mesh-classification method based on angular histograms

Tomasz Zawadzki
Florida Institute of Technology
150 W. University Blvd
USA, 32901, Melbourne, Florida
E-mail: tzawadzki@my.fit.edu

Slawomir Nikiel
University of Zielona Gora
Licealna 9
Poland, 65-417, Zielona Gora
E-mail:s.nikiel@issi.uz.zgora.pl

Eraldo Ribeiro
Florida Institute of Technology
150 W. University Blvd
USA, 32901, Melbourne, Florida
E-mail: eribeiro@cs.fit.edu

ABSTRACT

Methods for analyzing of 3-D mesh topology play an important role in computer graphics problems such as segmentation, correspondence, shape recognition, editing, and animation. We present a new approach for recognition, analysis, and classification of 3-D mesh-based objects. Here, we show how histogram descriptors based on the angles between vertices can be used for grouping similar shapes and for shape classification. Our method can be customized to classify objects of varying mesh complexity (i.e., level of detail) by changing the descriptor's resolution parameter. We demonstrate the effectiveness of our method on a 3-D object database generated using shape grammars and a hybrid morphing algorithm.

Keywords

Verification method, 3D mesh, recognition, histogram.

1. INTRODUCTION

Feature-based classification of three-dimensional-meshed objects allows for efficient and robust editing of complex CAD models by effectively representing shapes using shape descriptors [Hof98, She04]. A current trend is to try to make 3-D shape descriptors available not only for applications such as shape retrieval [Fun03, Liu10] but also for shape matching [Bus05]. Unlike its 2-D domain counterparts, 3-D shape description of mesh-based solids remains an open problem both in terms of feasibility and accuracy. Since databases containing 3-D objects have become more accessible to users with no previous CAD/CAM software experience, the need to develop new methods for 3-D search becomes highly relevant [Fun03, Bus05].

Usually, designers and manufacturing companies use semantic keywords, or meta-description to search for graphical 3-D objects in databases. Given a small set of initial objects that resemble the intended target, these objects are iteratively shaped (morphed) to render the final shape that the artist had in mind. In this paper, we introduce an alternative approach to meshed-object description and recognition. Our 3-D mesh classification method works in two phases. In the first phase, we set up a database of objects with different topologies and create average angular histograms for each class of meshed objects (i.e., caves, clouds and architecture). In the second phase, we generate random objects for which baseline histograms are created. After that, we recognition is achieved by comparing the generated baseline histograms with those that were created in the first stage and then we can check if objects are recognized correctly. Actually few independent research teams working on software development in project AIM@SHAPE [www]: MeshLab – editing of unstructured 3D triangular meshes, TriMesh2MT – convert mesh to multi-triangulation, AXEL – algebraic software-components for geometric modeling, COMREN – multi-resolution coding of mesh files.

2. OTHER WORKS

Methods for the analysis and classification of 3-D objects use a metric, which prescribes a distance function over the object's surface. Tasks for which shape metrics play a key role include shape decomposition, shape signatures representation, and surface parameterization. Well-known surface metrics include the geodesic [Car76] and the isophotic [Pot04] distances, where the latter measures angles between surface normals; a combination of the

two is also common [Lai07]. Anisotropic geodesic metric based on the curvature tensor has also been considered for parametric surfaces [Seo08]. Another metric receiving recent attention is the diffusion distance [deGoe08], which accounts for the degree of connectedness between two surface points.
Alternatively, one can derive a discrete distance function for mesh primitives by assigning attributes to primitives, defining edge weights between adjacent attributes, and finally computing distances over the shortest paths in the primal or dual graph of the mesh [Kat03].

Interesting recent approaches include: a part-aware surface metrics [Mar11] (Fig. 1) and cubeness measure [Liu09] (Fig. 2).

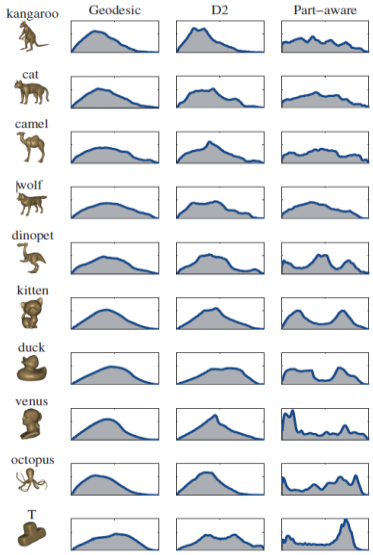


Figure1. The last three columns show the distance histograms of each model, obtained using geodesic, D2 and part-aware distances.

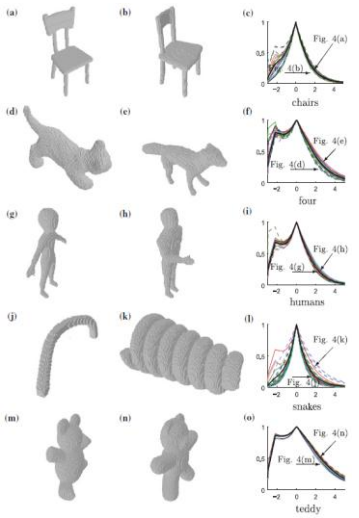


Figure 2. Each row presents the graphs GS(β) of all shapes from certain class (on the right), and two shapes whose GS(β) graphs differs

Another interesting approach is a Surflet-Pair-Relation Histograms [Wah03] based on a novel four-dimensional feature that describes the intrinsic geometrical relation between a pair of surflets, i.e., oriented surface points in 3D space. The statistical distribution of this feature as sampled from an object's surface captures both local and global aspects of shape. Empirically learned histograms of the feature distribution are demonstrated as a compact and efficient representation of arbitrary 3D shapes. This representation allows for rapid classification of shapes based on a single histogram per object model, independent of translation and rotation.

3. HISTOGRAMS

Our method represents shapes using a histogram of inter-vertex angles. We assume that models consist of edges, each of them determined by two points (i.e., vertices). In addition, the vertices are assigned normal vectors. We begin to build the representation by creating a set of edges of all triangles of the model. Here, edges belonging to two triangles are added twice, but perhaps with different normal vectors. Then, a value in the interval [0, 1] is calculated for each edge. This value is the difference of the two angles between normal vectors and the line containing the vertices (Fig. 3).

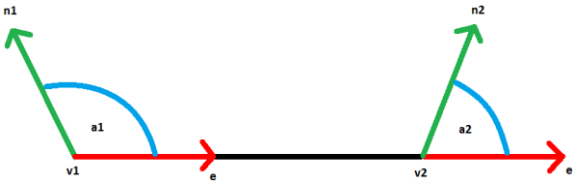


Figure 3. Calculation of the angles between the vertices.

$$e = \frac{(V_2 - V_1)}{|V_2 - V_1|} \quad (1)$$

$$a_1 = \arccos(e \bullet n_1) \quad (2)$$

$$a_2 = \arccos(e \bullet n_2) \quad (3)$$

$$d = a_2 - a_1 \quad (4)$$

$$v = \frac{d}{2(\rho+1)} \quad (5)$$

where: n_1 and n_2 are the normal vectors. For concave surfaces, this value belong to the interval [0,0.5). For convex surfaces, it lies the interval (0.5, 1].

In the next step, a discrete histogram is created indicating the frequency of numbers lying in the given interval. This is done as follows – for each value from the set interval histogram is calculated according to the formula:

 ISBN 978-80-86943-76-3

$$i = floor(v * R) + 1 \quad (6)$$

where: R is the number of histogram intervals.

Then the multiplicity of the i-th interval is incremented by one. After consider all values in the set, the histogram is normalized by dividing its compartments by the total number of edges. The normalization ensures that the sum of the ranges of the histogram equals to 1. For classification, it is necessary to prepare the histograms corresponding to a master object classes (Fig. 4.).
Verification of the correctness of the model depends on the density of the grid. We obtained the best results for $L_D=10$ (Fig. 5.).

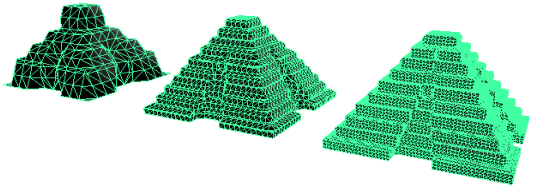


Figure 4. Level of detail (L_D), from left L_D=2, L_D=5, L_D=10.

When density is large, the histogram is reduced to a single bar representing the most common angle. The user can control the histogram resolution (R) to improve object classification. Figure 5 shows a set of histograms of an object for varying resolutions.

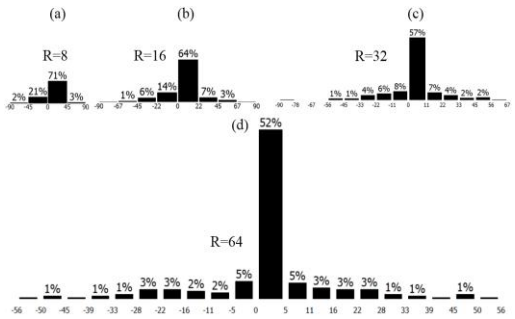


Figure 5. Varying resolutions for an object: (a) 8, (b) 16, (c) 32, (d) 64.

Creating a shape base

One stage of classification is to create a wide shape base that will be used for shape recognition.

From all objects selected to one class, we created average histograms for three main classes: architecture, clouds, and caves (Fig. 6).

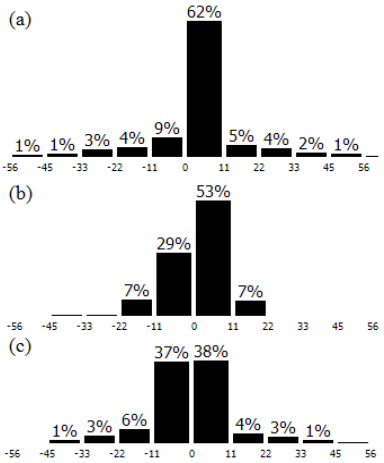


Figure 6. Main histogram classes: (a) architecture, (b) clouds, and (c) caves.

Our method produced different histograms for all of tested classes and in the next step it helps in recognition based on comparison of main classes with randomly generated objects using a calculation of Mean Squared Error (MSE).

4. RECOGNITION

We have tested our method using a database of 60 objects consisting of 20 caves, 20 clouds, and 20 architectural buildings. All objects were recognized correctly. Here, we are showing only a few selected objects from each class. Based on the average histograms, it was possible to recognize new objects (Figs. 7-9).

Figure 7. An example objects recognized as architecture.

 ISBN 978-80-86943-76-3

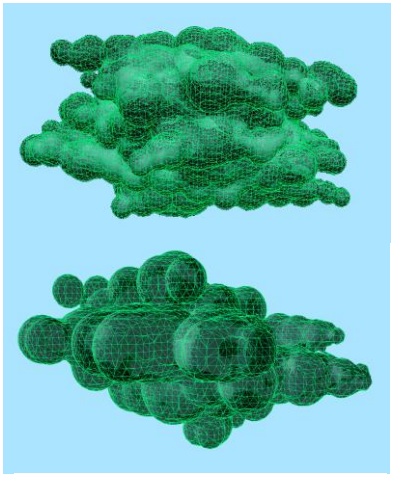


Figure 8. An example objects recognized as a clouds.

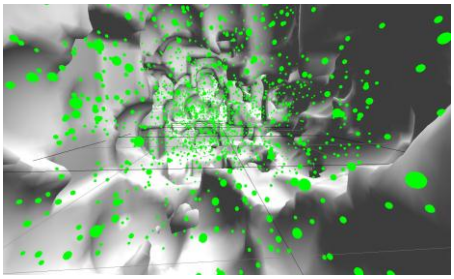


Figure 9. An example object recognized as a cave – view directly from editor.

5. SUMMARY

In this article new approach for analysis and classification of meshed models was presented. The method was evaluated on a database containing shapes generated by hybrid of shape grammar and morphing. 3D-meshes were created from functional description of implicit objects. That technique did not produce predictable results because generated set of points and triangles dependent only on mesh density – sampling.

The 3-D classification method based on histograms of the number of vertices and the angles between them. The method was able to group different models and then assign them to the pre-selected classes. The method achieved the best results for $L_D=10$ but it can be useful for every values of the L_D parameter. The metric can be used in identifying any three-dimensional shapes with hard-surfaces. In future work, we plan to extend the database and improve selection of angles for more detailed objects having most angles between (-10 and 10 degree) for a better classification.

6. REFERENCES

[Bus05] Bustos B., Keim D.A., Saupe D., Schreck T., Vrani´c D.V.:Feature-based similarity search in 3D object databases.ACMComput. Surv. 37(4), (2005), pp. 345–387.

[Car76] Carmo M. P. D.: Differential Geometry of Curves and Surfaces. Prentice Hall, 1976.

[deGoe08] De Goes F., Goldenstein S., Velho L.: A hierarchical segmentation of articulated bodies. Computer GraphicsForum (Special Issue of Symposium on Geometry Processing) 27, 5 (2008), pp. 1349–1356.

[Fun03] Funkhouser T., Min P., Kazhdan M., Chen J., Halderman A., Dobkin D., Jacobs D.: A search engine for 3D models. ACM Trans. Graphics, 22(1), (2003),pp. 83–105.

[Hof98] Hoffmann C., Joan-Arinyo R. : On user defined features, Computer Aided Design 30 (1998) pp. 321–332.

[Kat03] Katz S., Tal A.: Hierarchical mesh decomposition using fuzzy clustering and cuts. ACM Trans. on Graphics 22, 3 (2003), pp. 954–961.

[Lai07] Lai Y.-K., Zhou Q.-Y., Hu S.-M., Wallner J., Pottmann H.: Robust feature classification and editing. IEEE Trans. Vis. & Comput. Graphics 13, 1 (2007), pp. 34–45.

[Liu09] Liu R., Zhang H., Shamir A., Cohen-Or D.: A Part-aware Surface Metric for Shape Analysis. Machine division and application, 2009, DOI: 10.1111/j.1467-8659.2009.01379.x, pp. 397-406.

[Liu10] Liu Y., Wang X.-L., Wang H.-Y., Zha H., Qin H.: Learning robust similarity measures for 3D partial shape retrieval. Int. J. Comput. Vis. 89(2/3), (2010), pp. 408-431.

[Mar11] Martinez-Ortiz C., Zunic J.: A family of cubeness measures. Machine division and application, 2011, DOI: 10.1007/s00138-011-0328-x,pp. 1871-1881.

[Pot04] Pottmann H., Hofer M., Haider C., Hanbury A.: The isophotic metric and its application to feature sensitive morphology on surfaces. In Proc. Euro. Conf. on Comp. Vis. (2004), pp. 560–572.

[Seo08] Seong J.-K., Jeong W.-K., Cohen E.: Anisotropic geodesic distance computation for parametric surfaces. In Proc.IEEE Int. Conf. on Shape Modeling and Applications (2008), pp. 179–186.

[She04] Sheffer A., Gotsman C., Dyn N.: Robust spherical parameterization of triangular meshes, Computing 72 (2004), pp. 185–193.

[Wah03] Wahl E., Hillenbrand U., Hirzinger G.: Surflet-Pair-Relation Histograms: A Statistical 3D-Shape Representation for Rapid Classification, Forth International Conference on 3-D Digital Imaging and Modeling (3DIM 2003) 2003, Banff, Alberta, Canada, IEEE Computer Society Press, pp. 474-481.

[Www] http://www.aimatshape.net

An Intelligent Flower Analyzing System for Medicinal Plants

Ranga Rodrigo
School of Electrical and
Computer Engineering
University of Moratuwa
Email: ranga@uom.lk

Kalani Samarawickrame
Informatics Institute
of Technology
Email: ksamarawickrema@gmail.com

Sheron Mindya
Informatics Institute
of Technology
Email: smindya@gmail.com

***Abstract*—The natural sciences have been transformed with the incorporation of advanced technologies, and the current technological wave of change is also revolutionizing biological science. With the wide use of herbal medicines in traditional medical systems, the demand for medicinal plants has increased enormously. Though the current trend has made the use of medicinal plants more popular, we still require better methods to distinguish between medicinal plants and plants which do not possess medicinal value. Manual methods of plant recognition rely on a plants flower information. However, these manual processes of flower recognition are not straightforward for lay persons unless expert guidance is provided. Motivated by this reasoning, an intelligent flower analyzing system has been developed to recognize medicinal plants. Various tests have been performed on 160 images taken from four types of flowers, to recognize flowers based on their colour and shape features. The experiment was carried out using Support Vector Machine (SVM) and Principal Component Analysis (PCA) methodologies respectively for colour and shape extraction. Acquiring an average of 65% recognition rate implies that the applicability PCA and SVM in the specified domain is a valuable step forward.**

I. Introduction

The demand for medicinal plants has increased over the past two decades as herbal medicines are being used widely in traditional medical systems in developed and developing countries [4]. Traditional herbal medicines such as ayurveda, homeopathy, naturopathy, unani are used in traditional medicine systems such as in China, Africa and Latin America. However, although the current trend has drawn our attention towards medicinal plants, people who do not have an in-depth knowledge in biology lack awareness in distinguishing an authentic medicinal plant from an adulterant. Wrong identification of medicinal plants is one of the factors that make herbal remedy unsafe. [1] Owing to the ignorance of the exact identities of plants used in ayurvedic practice, many exotics are being used mistakenly or as substitutes in the absence of the plants originally recommended. Plants are classified most often on reproductive (flower) characteristics as opposed to vegetative (leaf, stem) characteristics. This has become customary since reproductive parts (petals, sepals, stamens and pistils) remain relatively unchanged over diverse environments, whereas vegetative parts tend to change depending on the environment in which they grow. In addition, flower recognition done with the use of flower guides is not straightforward for an amateur [5]. Identifying a flower using a field guide or key without expert guidance is also a time-consuming task. Furthermore, the fact that some of the flowers being relatively similar and different examples of the same flower differ in colour and shape implies that the recognition by laypersons or pattern recognition systems is not straightforward. Flower recognition becomes cumbersome when using keys as they require answers for a series of questions in order to recognize flowers. These features often relate to the internal structure of the flower which is in most cases visible only when it is dissected [9]. Different approaches have been proposed concerned with recognition of characters, human faces, cancer from chest X-ray, etc. However, outfield living objects such as fish, animals, and plants have not attracted much attention [6]. Moreover, the large similarities between flowers results in an extra challenge in flower recognition over categories such as bikes, cars and cats. In addition, flowers are non-rigid objects that can deform in many ways, and consequently there is also a large variation within classes [9]. Different approaches have been proposed for flower recognition. These approaches are based on either colour or shape information associated with the flower image. Researchers have presented an interactive extraction system for flower recognition based on colour [3] and conversely another group has proposed a model-based interactive flower recognition system based on the shape of flowers [17]. Based on [11] researchers have also proposed a method of extracting flower regions based on Intelligent Scissors. However, these models are highly user dependant as the systems require the user to adjust the model according to the shape of flower in order to obtain more accurate results. The automatic boundary extraction method proposed by Saitoh et al. [10], solves the aforementioned issue to some extend thus outperforms the Intelligent Scissors approach. Moreover the research shows that using colour or shape feature alone, is insufficient for flower recognition. In this paper, a novel approach has been introduced by the authors to incorporate colour information with shape of flowers. The technique proposed in this paper for flower recognition is composed of the following steps. Initially, the flower recognition system has been described briefly. The process of shape recognition based on Principal Component Analysis(PCA) is described in the following section followed by the use of Support Vector Machine for classification using

 ISBN 978-80-86943-76-3

colour information. Afterwards an outline of the experiment is carried out and the test results have been stated. Finally the paper is concluded by stating the findings and the overall outcome of the proposed approach.

II. System Description

A database of 160 flower images from four different types of flowers has been used in the system. These images have been obtained under various conditions. All pictures have been captured using a digital camera having a constant dark background, lighting condition and a constant distance from the camera to the flower. These are 75 by 50 pixels. From the 160 images, 120 have been taken for the training database and the rest have been used as test images. When the system is given an image of an unknown flower, it extracts colour and shape information and selects the respective flower from the training database. Afterwards, the particular medicinal plant details along with an image of the medicinal plant are provided by the system. With the use of these images the colour and shape information were obtained as illustrated in the next sections.

III. PCA Approach for Shape Extraction

In flower recognition, the flower image is transformed into a small set of characteristics, called eigenfaces, which are the principal components of the initial training set of face images. Similarly, it can be defined as the mean squared distance between the data points and their projections [2].This is called the eigenface method. It is a well known template matching approach where the feature values transformed by principal component analysis are used for recognition [6]. In order to process eigenfaces calculation, each two dimensional (NxN) image in a set of training images, is converted into a column vector of dimension NxN. Subsequently, all columns are joined in order to compute the average image flower vector. This is followed by obtaining mean flower and co-variance matrix to generate new eigenvectors [7]. Eigenvectors are a set of features which together characterize the variation between images. Each image location contributes more or less to each eigenvector [14]. Thus the eigenvectors can be displayed as ghostly faces of flower images. Nevertheless, aforementioned vector contains information on each pixel after projecting the image into eigenface components. When a new image is given, it is also represented by its vector of weights. Hence, recognition of the test image is done by locating the image in the database which has the closest weight to the test image [16]. An image vector will contain w*h pixels of information when width and height contain w and h number of pixels respectively. However, prior to shape extraction the images were enhanced by compressing images and filtering noise in order to reduce execution time and preserve edges of the flowers respectively.

IV. Recognition using Euclidean Distance

For an unknown flower image, the closest matching train image is selected with the use of Euclidean Distance. The distance from the test image to each eigenface which is obtained subtracting the average flower image vector from each training image is calculated using Euclidean Distance. Thus it gives the nearest image from the training data set which matches with that of the given flower image. Based on the calculated minimum Euclidean distance, the relevant flower index from the training database is selected. Afterwards, the matched flower image along with the relevant medicinal plant and the description is provided by the proposed flower analyzing system.

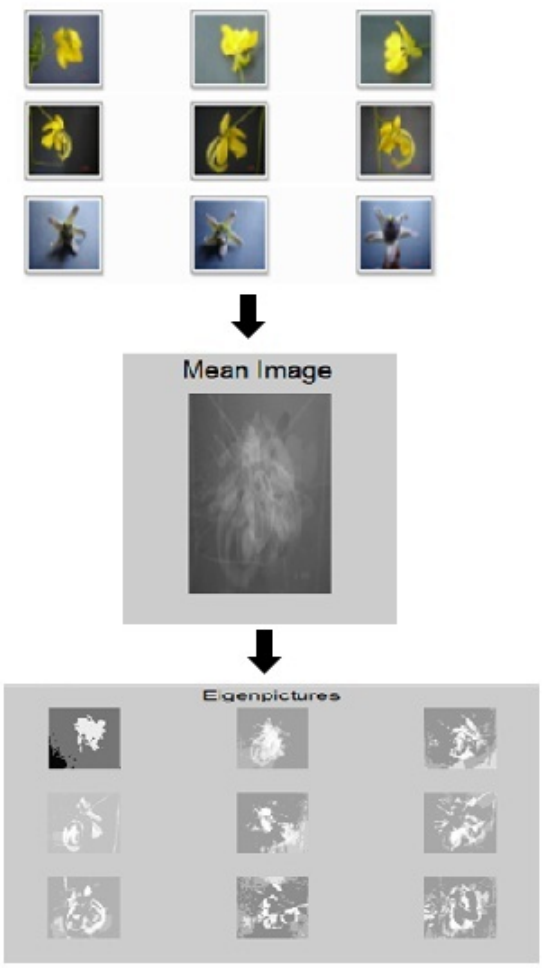


Fig. 1. Generating Eigenpictures

V. Machine Learning for Recognition using color

Support Vector Machines are based on the results of statistical learning theory carried out by Vapnik. SVM maps feature vectors into a higher dimensional space and classify data using linear algebra by employing a kernel function. Then an optimal hyper plane that fits into the training data is created. In a linear classification the margin between the separating hyper plane and the nearest feature vectors from both classes is maximal. The feature vectors closest to the hyper plane are

called support vectors. SVM has evolved from sound theory to implementation and experiments while Neural Networks has followed a more heuristic path, from application and extensive experimentation to theory. Moreover it is stated that SVM has achieved practical learning benchmarks in digit recognition, computer vision and text categorization [15]. Thus SVM had been adopted as the machine learning technique in order to classify medicinal plants based on flowers using the color feature.

The color feature plays a foremost role when classifying a flower image. According to [13] , a color space with a separation of the luminance and chrominance information tends to provide better results than a color space with these information mixed (as RGB). Luminance and chrominance are separate in the LAB color space which is more effective for this task. a and b dimensions of Lab color space had been applied in order to extract color information. L has not been considered since it refers to the lightness in the image. Dimension a denotes the position between red/magenta and green where as dimension b denotes its position between yellow and blue.

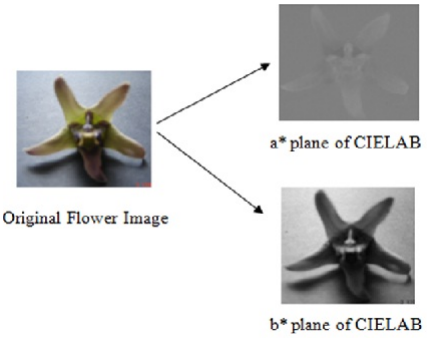


Fig. 2. Seperation of a and b color spaces

Initially a and b color planes have been extracted from a given image and combined before fed into the SVM classifier. At that moment it is also labeled with the relevant plant name. A train data set will be created from the described mechanism. Hence, when a test flower image is given, a and b color features would be extracted and combined in order to be used for testing. A linear classification would be performed and whether the flower falls into a particular category would be decided based on which side of the hyperplane, the test image lies.

VI. Evaluation

The system had been evaluated by considering classification performance using both shape and color features for 4 kinds of medicinal plants. As illustrated in the tables below, testing had been done for 10 flowers from each kind.Confusion Matrix had been drawn in order to denote how many flowers of a particular category had been identified correctly and how many flowers had been identified falsely. Analysing the data in Table 1 it could be seen that Shape Extraction using PCA has given an overall accuracy of 77.5%. According to Table2 classification of flowers using color information using SVM has given an overall accuracy of 82.5%. Based on Table2 when using SVM three instances had been identified where the Wara flower had not fallen into any of the 4 categories. The system had also been tested combining the above mentioned technologies. Thus the results of Table 3 shows that SVM and PCA technologies combined has given an overall accuracy of 65%. In this mechanism an output was given only if both PCA and SVM identified the flower to be in the same category. Accuracy can be enhanced by capturing images with a higher resolution and increasing the number of training and testing images.

TABLE I
Confusion Matrix for classification of flowers using PCA

	Clitoria	Ranawara	Ahala	Wara
Clitoria	8			
Ranawara		8	4	
Ahala	2		6	1
Wara		2		9
Accuracy	80%	80%	60%	90%

TABLE II
Confusion Matrix for Classification using SVM

	Clitoria	Ranawara	Ahala	Wara
Clitoria	10			
Ranawara		6		
Ahala		3	10	
Wara		1		7
Accuracy	100%	60%	100%	70%

TABLE III
Confusion Matrix for classification of flowers using PCA and SVM

	Clitoria	Ranawara	Ahala	Wara
Clitoria	8			
Ranawara		6		
Ahala			6	
Wara				6
Accuracy	80%	60%	60%	60%

VII. Conclusion and Discussion

This application is highly important in the development of natural sciences especially in plant taxonomy related studies as the manual recognition done using books is time consuming and the use of internet to search for information would not be accurate all the time. Furthermore at the exportation of flora, this kind of system would be useful in order to prevent bio piracy as it would be straightforward and practical than expecting the lab to test samples of each and every flora that is been exported.

In this paper, a generic approach was presented in order to recognize medicinal plants using its flower. The shape and color features of a flower had been taken into consideration when determining the plant. The results of using Support Vector Machine for color classifying a flower using color information indicate the optimality of the technique for the task. It also shows that incorporating shape information has also contributed for the high accuracy rates. In order to improve the recognition rate of 65%, leaf recognition could also be incorporated as discussed in [8].

It is believed that a system like this would contribute to the Sri Lankan ayurvedic treatments. At the same time since the solution has a generic approach where it could be trained to recognize any kind of flower and not restricted to medicinal plant flowers, the system could be used in many applicable areas where flower recognition is needed.

REFERENCES

[1] D Bagozzi. (2004, Feb) Medicinal plants guidelines to promote patient safety and plant conservation for a US$ 60 billion industry. [Online]. http://www.who.int/mediacentre/news/notes/2004/np3/en/

[2] C Bishop , Pattern Recognition and Machine Learning. Singapore: Springer Science+Business Media., 2006.

[3] M Das, R Manmatha, and E. M Riseman, "Indexing flower patent images using domain knowledge," Intelligent Systems and their Applications, vol. 14, pp. 24-33, Sep/Oct 1999.

[4] D. Deshpande, A Handbook of Medicinal Herbs. India: Agrobios, 2006.

[5] A. Evans , J. Sikorski , P. Thomas , J. Zou, and G. Nagy ,"Computer assisted visual interactive recognition (CAVIAR) technology," in 2005 IEEE International Conference on Electro Information Technology, NY, 2005, pp. 6-6.

[6] K Hyun-Chul, K Daijin , and S. Y Bang, "Face recognition using the mixture-of-eigenfaces method," Pattern Recognition Letters, vol. 23, no. 13, pp. 1549-1558, November 2002.

[7] A S Khan and L K Alizai, "Introduction to Face Detection Using Eigenfaces," in ICET '06. International Conference on Emerging Technologies, 2006, pp. 128 - 132.

[8] N Kumar, P N Belhumeur, A Biswas, David W. Jacobs, W. John Kress, Ida C. Lopez, Joo V. B. Soares, "Leafsnap: A Computer Vision System for Automatic Plant Species Identification", ECCV Proceedings of the 12th European conference on Computer Vision, 2012, vol.2, pp. 502 - 516.

[9] M Nilsback and A Zisserman, "A Visual Vocabulary for Flower Classification," in 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2006, pp. 1447-1454.

[10] T Saitoh,K Aoki, and Kaneko, "Automatic recognition of blooming flowers ," in Proceedings of the 17th International Conference on Pattern Recognition , 2004, pp. 27-30.

[11] T Saitoh and T Kaneko,"Automatic recognition of wild flowers ," in 15th International Conference on Pattern Recognition, 2000, pp. 507 - 510 vol.2.

[12] M Storer , M Roth Peter , M Urschler, and B Horst , "Fast-Robust PCA," in Proceedings of the 16th Scandinavian Conference on Image Analysis , 2009, pp. 430-439.

[13] L Torres, J Y Reutter, and L Lorente , "The importance of the color information in face recognition," in International Conference on Image Processing, Kobe, 1999, pp. 627 - 631.

[14] M A Turk and A P Pentland, "Face recognition using eigenfaces," in IEEE Computer Society Conference on Computer Vision and Pattern Recognition , Maui, HI, 1991, pp. 586 - 591.

[15] L Wang, "Support Vector Machines: Theory and ApplicationsSeries," Studies in Fuzziness and Soft Computing, vol. 177, 2005.

[16] W Zhao, R Chellappa, P J Phillips, and A Rosenfeld, "Face recognition: A literature survey," ACM Computing Surveys (CSUR), vol. Volume 35, no. Issue 4, pp. 399 - 458, December 2003.

[17] J Zou and G Nagy, "Evaluation of model-based interactive flower recognition," in Proceedings of the 17th International Conference on Pattern Recognition, 2004, pp. 311 - 314 Vol.2.

Light Field Retrieval In Compressed Domain

[(a)]Lakhdar BELHALLOUCHE [(a)]Kamel BELLOULATA [(b)]Kidiyo KPALMA
Lakhdar_belhallouche@hotmail.fr Kamel.Belloulata@USherbrooke.ca kidiyo.kpalma@insa-rennes.fr
[(a)]Université Djillali Liabés, Algeria [(b)]UEB INSA, IETR, UMR 6164, F-35708 RENNES, France

ABSTRACT

In this paper, we present a novel approach for light field retrieval in the transformed domain. The light field data set is characterized with low wavelet transform coefficients. Our algorithm first applies a wavelet transform on 20 key images of the light field structure. The low wavelet transform coefficients are then quadtree decomposed, into a set of homogeneous blocks. Each set of homogeneous blocks represent an object in the considered image. We use texture and color features to characterize the object image; the similarity between two images is measured by matching their histograms. The experimental results and comparisons show the performance of the proposed technique.

Keywords

light field, retrieval, similarity measures, wavelet analysis, quadtree.

1. INTRODUCTION

The rapid advance of digital technologies has improved the methods of acquisition and rendering of 3D models. We can see today that the databases of 3D objects are present in many areas: games, multimedia, medical applications, cultural heritage.

The ease of acquisition and reconstruction of 3D models allows large databases creation, so it becomes difficult to navigate and find information. Indexing 3D objects thus appears as a necessary and promising solution to manage this new data type.

Current 3D models search engines use 2D shapes drawn by user as a query image. The objective is to find relevant 3D models efficiently and correctly by querying from a database models. In other words, we search the most similar model of the database to the query [1]. The proposed approach in [2] is based on 20 silhouettes characterization. This method represents the object in 20 different locations on the sphere. The 20 views are used to construct a combined texture and color feature, which characterizes the captured 3D object. Chen et al. [1] propose to match 3D objects based on view similarity measure. Other view based retrieval methods use 3D models matching according to geometric distribution, topology structure and curvature of a patch [3].

These approaches achieve good retrieval scores, however higher dimension points make the analysis more difficult. Image-based retrieval methods try to address this problem by measure the similarity between rendered projections. Ohbuchi et al., for instance, propose a characterizing scheme of each view image using the SIFT algorithm [4]. Local features were then integrated into a histogram using a bag of features approach for retrieval.

Light field is a structure-based image representation for interactive visualization of any new point of view leading to high data volume. To reduce the cost of storage of light fields, one can combine the two operations of transformation and indexing to achieve retrieval directly in the transformed domain.

In this paper, we propose a light-field-based image retrieval in the wavelet domain. This system uses an approximated light field representation as proposed in [2], which considers only 20 captured key views from a fixed set of positions on the sphere. Each view object is firstly converted into YC_bC_r space and then each image component Y, C_b, C_r is transformed using wavelet. The resulting low frequency coefficients are decomposed into homogeneous blocks by quadtree. We finally construct texture and color features from the low-pass band of the wavelet transform. These features are organized into a histogram to describe the object. At the end, the retrieval step is based on the similarity measure using Euclidean distance between the query view and candidate views.

The rest of this paper is organized as follows. We first describe the acquisition system [4]. Next we describe our method for conversing and transforming light fields. We then explain the texture and color

features construction. Finally, we present and analyze the results from our experiments and provide a conclusion.

2. Data acquisition system

The acquisition system (Figure 1) used in [5] takes in consideration all rotations, scaling and moving of the object.

Ameshin [5] proved that this acquisition system of light fields offers the best possible correlation alignment between the models. So our indexing system, to be efficient, must be robust to rotation, displacement and scale change.

In our case, we use an image data base [7] acquired in accordance with the system described above.

We then select only 20 views with 18° rotation angle between two consecutive cameras. Obtained views represent the light field of the object.

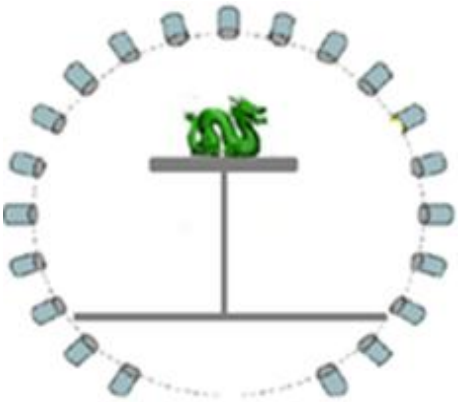


Figure 1. Acquisition system

3. The proposed scheme

The proposed approach consists of two steps: firstly, we use wavelet transformation and secondly, we apply quadtree segmentation onto the low frequency coefficients.

3.1. Wavelet transformation

First, color image is converted from RGB to YCbCr and the image components (Y, Cb, Cr) are transformed using wavelet decomposition based on the biorthogonal Cohen–Daubechies–Feauveau 9/7 (CDF 7/9) filters.

Each component is decomposed up to 2 levels.As shown in Figure 2, each image is converted into 3 components YCbCr.

For each image component transformed using DWT CDF 9/7, we consider only low wavelet coefficients, for image characterization with texture and grey level histograms.

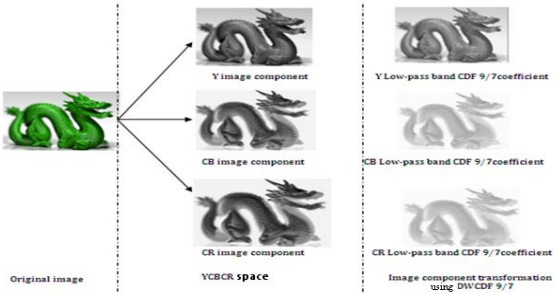


Figure 2. View transformation using CDF 9/7 wavelet transform

3.2. Image quadtree decomposition

Each low frequency sub-band is successively decomposed into quadrant depending on the complexity of the coefficients. The sub images are then iteratively decomposed until all blocks are homogenous (figure 3): the standard deviation of the coefficients within the block is below a predefined threshold. A sub image is considered to be homogeneous block if its standard deviation is lower than the threshold.

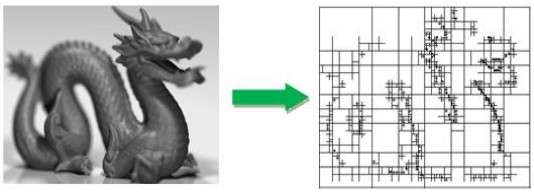


Figure 3. Image quadtree decomposition

4. Light field view indexing

Based on the quadtree-decomposed sub-bands, texture features and color features are extracted.

4.1. Texture feature construction

Firstly, for each image block of each component, the co-occurrence matrix is constructed. This matrix characterizes the texture of each image block by calculating how often pairs of pixels, with specific values and in a specified spatial relationship, occur in a block [9]. From this matrix, we calculate the following statistical measures (figure 4) to be texture features:

Energy feature: is the squared sum of the values of the co-occurrence matrix.

$$\sum_{i,j} P(i,j)^2 \quad (1)$$

Contrast feature: represents the contrast distance between two matrix elements.

$$\sum_{i,j} |i-j|^2 P(i,j) \quad (2)$$

Homogeneity feature: measures the distribution of the co-occurrence matrix elements around the diagonal.

$$\sum_{i,j} \frac{P(i,j)}{1+|i-j|} \quad (3)$$

 ISBN 978-80-86943-76-3

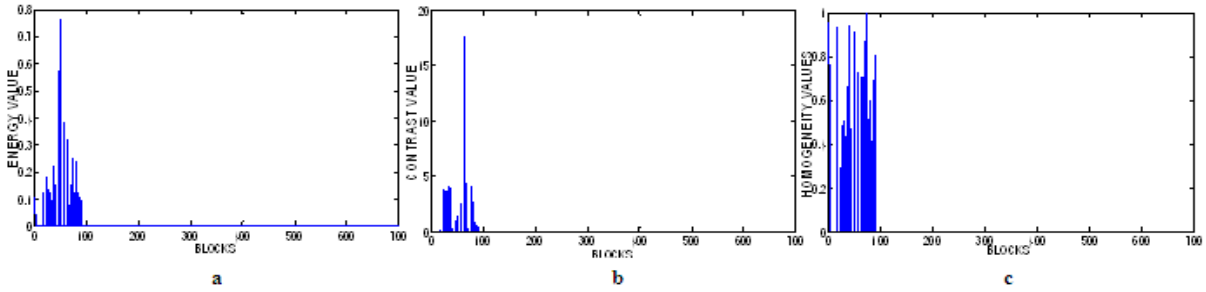


Figure 4: examples of texture image features: (a) energy feature, (b) contrast feature, (c) homogeneity feature.

Now, each image block is characterized by the mean values of its energy, contrast and homogeneity from the corresponding blocks in the three components (Y, C_b, C_r):

$$Energy=(Y\_E+C_b\_E+C_r\_E)/3 \quad (4)$$

$$Contrast=(Y\_C+C_b\_C+C_r\_C)/3 \quad (5)$$

$$Homogeneity=(Y\_H+C_b\_H+ C_r\_H)/3 \quad (6)$$

where x_E, x_C and x_H are the Energy, the Contrast and the Homogeneity values respectively from the x component. Figure 4 shows an example of these textures parameters.

4.2. Color feature construction

The second step of our indexing consists of color vectors construction using the mean value of each block of the image components.

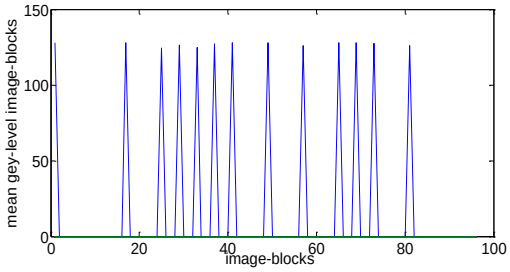


Figure 5. Example of color feature

Figure 5 represents the color image feature, which is the mean of grey level values in the domain Y, C_b, C_r respectively:

$$Color = (Y\_gl + C_b\_gl + C_r\_gl)/3 \quad (7).$$

where Y_gl, C_b_gl and C_r_gl are the gray-level values of the image blocks in the respective components Y, C_b and C_r.

Once all the features have been extracted, every homogeneous block is represented by a four-element vector as <Gray_level, Energy, Contrast, and Homogeneity> (Figure 6).

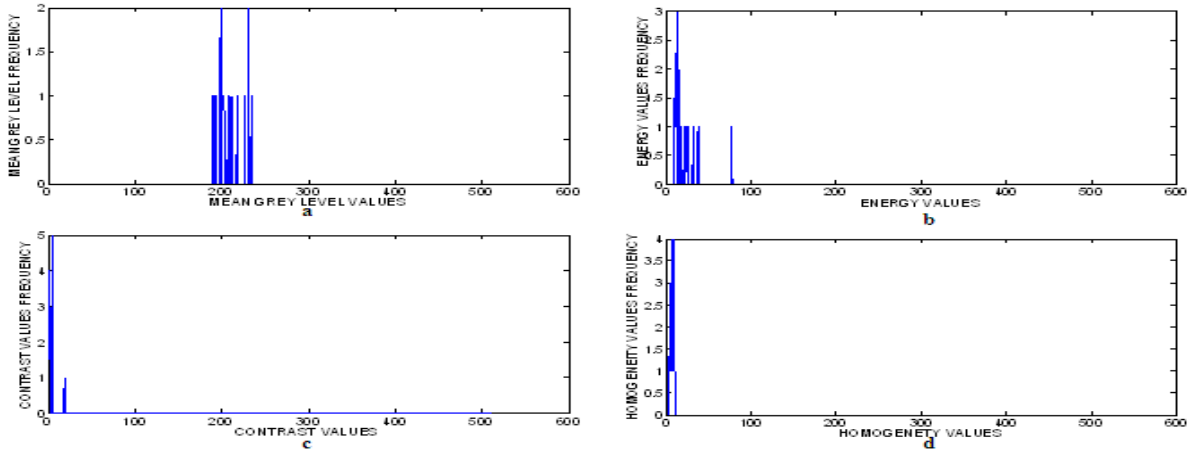
4.3. Histogram generation

The histogram of the Energy feature represents the frequencies of all the texture energy values. This histogram is then quantized into M bins such that:

$$H\_E= \{h(b_1), h(b_2) \ldots h(b_M)\} \quad (8)$$

With M=32 bins and where $h(b_i)$ is the frequency of the texture energy value at bin b_i.

The same method is used to construct the contrast, homogeneity and color histograms H_C, H_Ho, and H_Cl, respectively. Figure 6 clearly illustrates that there is a few histogram values that appear in high frequency in the histograms. This figure shows that the first bins are the dominant ones in the histogram [8]. To reduce the length of the histogram features, only high frequency 32 first bins are selected to form a reduced histogram. As stated in [8], such representations can yield better image characterizing.



 ISBN 978-80-86943-76-3

Figure 6. Color (a), Energy(b), contrast(c) and homogeneity(d) histograms

4.4. Similarity Measurements

As many of current Retrieval approaches, our similarity measurement is based on the Euclidian distance of the histograms:

$$D = \sqrt{\sum_{i=1}^{n} (HCi - HQi)^2} \quad (9)$$

Where H_C and H_Q are the feature histograms of candidate image C and Query image Q respectively with size n.

5. Experimental Results

The proposed algorithm is tested on a database of images. The database consists of 100 images having 5 image categories with 20 vies each. All images are in the RGB space. In order to measure retrieval effectiveness for our image retrieval system, we use the precision and recall values. We select three different images of each category as query images.

The resulting Euclidian distances between the query image feature histogram and each feature histogram in the database, are used to evaluate our retrieval system.

$$\text{Precision} = \frac{\text{number of relevant images retrieved}}{\text{total of images retrieved}} \quad (10)$$

$$\text{Recall} = \frac{\text{number of relevant images retrieved}}{\text{total number of relevant images in database}} \quad (11)$$

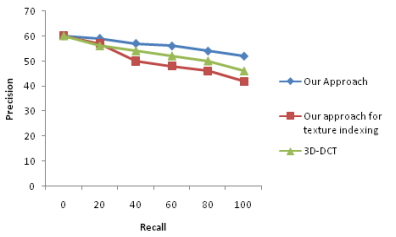


Figure 9. Our retrieval method evaluation

A number of experiments were performed to evaluate the performance of our proposed approach and its variants. The upper graphs show the retrieved results using our approach and testing our approach with texture database respectively. The retrieved results are ranked in the top ten in similarity. It can be found that our method would perform better than the 3D-DCT method in terms of the precision rate.

6. CONCLUSION

In this work we have addressed the problem of light field access by the constitution of an indexing system and image retrieval in the transformed domain. The proposed solutions were evaluated in the context of image based light field retrieval. These solutions allow to reduce the comutational complexity and the discriptiors length.

Acknowledgements: This work was partially supported by the Franco-Algerian cooperation program PHC TASSILI grants. Entitled "Système Conjoint de Compression et d'Indexation Basé-Objet pour la Vidéo (SCCIBOV)", the project gathers members of INSA de Rennes, UTC, the University of SidiBel Abbès and the University of Mascara.

REFERENCES

[1] Yu-TeShen, Ding-Yun Chen, Xiao-Pei Tian and Ming Ouhyoung 3D Model Search Engine Based on Lightfield Descriptors EUROGRAPHICS 2003 / J. Flores and P. Cano

[2] Y.-T. Shen D.-Y.Chen, X.-P.Tian and M. Ouhyoung. On visual similarity based 3D model retrieval. In Eurographics, 2003.

[3] L. Cieplinski, M. Kim, J.-R. Ohm, M. Pickering and A. Yamada, "Text of ISO/IEC 15938- 3/FCD Information Technology - Multimedia Content Description Interface - Part 3 Visual", ISO/IEC JTC1/SC29/WG11/N4062, Singapore, Mar. 2001.

[4] R. Ohbuchi, K. Osada, T. Furuya, and T. Banno. Salient local visual features for shape-based 3D model retrieval. In Shape Modeling International, 2008.

[5] Ameesh Makadia and Kostas Daniilidis, Light Field Similarity for Model Retrieval EUROGRAPHICS 2003 / P. Brunet and D. Fellner

[6] Shou-Yi Tseng, Zhi-Yu Yang, Wen-Hsuan Huang, Chin-Yi Liu and Yi-Huei Lin Object Feature Extraction for Image Retrieval Based on Quadtree Segmented Blocks 2009 World Congress on Computer Science and Information Engineering.

[7] http://web.media.mit.edu/~gordonw/SyntheticLightFields/

[8] Chuo-Ling Chang, Xiaoqing Zhu, Prashant Ramanathan, and Bernd Girod, Light Field Compression Using Disparity-Compensated Lifting and Shape Adaptation IEEE transactions on image processing, vol. 15, no. 4, april 2006.

[9] Haralick, R.M., and L.G. Shapiro. Computer and Robot Vision: Vol. 1, Addison-Wesley, 1992, p. 459

[10] Cong BAI, Kidiyo KPALMA and Joseph RONSIN Color Textured Image Retrieval By Combining Texture And Color Features "European Signal Processing Conference (EUSIPCO-2012), Bucharest : Romania (2012)".

Comparison of segmentation evaluation methods

Štěpán Šrubař
FEECS VSB-TUO
17. listopadu 15
708 33, Ostrava, CS
stepan.srubar@vsb.cz

ABSTRACT

Quality of segmentation depends on evaluation method we use. In case of specific images we use specific ways how to define quality. In general cases we have to use common methods. They differ in complexity as well as in quality. This article presents some of these methods and shows comparison of their quality on large image set.

Keywords

Segmentation, evaluation.

1 INTRODUCTION

Segments in segmentation can be measured for some specific properties (perimeter, area, curvature) or we can process or modify image information in each segment separately. For perfect results, we need high quality segmentation which is often created by a segmentation algorithm. Therefore, quality of the algorithm should be measured by an evaluation method and its quality should be evaluated as well.

2 EVALUATION METHODS

This article includes over 30 evaluation methods. Some methods needed small modification for evaluation of segmentations. Methods are divided into 6 categories which are provided in the following subsections.

2.1 Segment and Intersection Size Based Methods

Multi-class Error type I and type II by W. A. Yasnoff et al. [YMB77] use intersection of segments. Final methods SM_1 and SM_2 are just weighted sums. Pal and Bhandari used difference and ratio of size of segments in their Symmetric Divergence (SYD) [PB93]. D. Martin et al. [Mar02] used relative size of intersection in Global (GCE), Local (LCE) and Bidirectional Consistency Error (BCE). I propose Global Bidirectional Consistency Error ($GBCE$) which comes out of GCE and BCE. Larsen in L sums maxima of relative intersection sizes [LA99]. Q. Huang et al. in normalized Hamming Distance (HD) used sizes of intersections [HD95]. Van Dongen proposed metric using maximal intersections (VD) [VD00]. Meilă and Heckerman used sizes of intersections for predefined correspondence of segments (MH) [MH01]. Nearly the same was proposed later by Cardoso and Corte-Real [dSCCR05] as Partition Distance (PD).

2.2 Methods Using Distance Measure

Yasnoff et al. compute squares of distances of pixels to corresponding segments (YD) [YMB77]. Strasters and Gerbrands used the same approach in F but they normalize it by number of mis-classified pixels and a parameter [SG91]. Figure of Merit by Pratt [Pra78] uses similar evaluation expression but evaluates border pixels. Necessary natural extension is presented here as $SFOM$. Monteiro and Campilho combine linear and logarithmic distance (MC) [MC06]. Paumard's Censored Hausdorff Distance was also extended ($SCHD$) [Pau97]. It is based on Hausdorff distance used for sets. Q. Huang and B. Dom proposed more methods but only one which uses average of distances is well defined and usable ($H_{2\mu}$) [HD95] . Segmentation Difference (SD) method computes discrepancy in number of segments and distance of pixels [Sru10, Sv11, Sru11]. For evaluation only distance is used. There are four alternatives of SD denoted by index.

2.3 Methods Based on Counting of Couples

If you take two random pixels, they could lie in the same segment or in different segments in the first or the second segmentation. Using combination of these properties we get four types of couples. Number of couples of each type is used in following methods: JC by Jaccard [BHEG02], FM by Fowlkes and

 ISBN 978-80-86943-76-3

Mallows [FM83], W_I by Wallace [Wal83] and M by Mirkin [Mir96] (which is even a metric) and Rand index [Ran71]. The latter method was extended by probability of pixels (*PRI*) [UPH05].

2.4 Methods with Statistical Approach

Yasnoff and Bacus proposed Object Count Agreement (*OCA*) which uses number of segments and sets of segments [YB84]. Entropy of segments is used in Normalized Mutual Information (*NMI*) [SGM00]. Another metric - Variation of Information (*VI*) was proposed by Meilă in [Mei03].

2.5 Graph Theory Methods

Method based on graph theory was proposed by Jiang et al. in [JMIB06]. Segments represent nodes in graph while their correspondences are vertexes. Result is sum of all vertexes in its maximum-weight bipartite graph.

2.6 Other Methods

Last method Fragmentation (*FRAG*) [SG91] uses difference of number of segments. Result can be tuned by two parameters.

3 METHODOLOGY OF COMPARISON

All implemented methods are practically tested on image data set consisting of pictures taken by a typical camera [MFTM01]. They consist of sample images and their ground truth segmentations.

Comparison of two segmentations results in a number. In each comparison we know if these segmentations belong to the same image or not. According to that property results should split into two clusters (same vs. different segmentations). Typically, result of segmentations belonging to the same image are low while results from segmentations from different images are high. Therefore, we could find optimal threshold to separate results into the two categories. Still, there could be results on the wrong side of the threshold. We will divide their number by size of the whole cluster and call them false acceptance (FA) or false rejection (FR) whether they lie on the one or the other side of the threshold. The threshold can be different for each method.

4 RESULTS

Methods were implemented and measured. Symmetric methods evaluated 3138496 couples of segmentations, while each asymmetric method processed 6276992 couples of segmentations. Results are presented in figure 1. All methods reaching 50% are practically unusable. From the results we could see that local refinement (*LCE*, *SD*) is better than intolerance to refinement (*BCE*) which is better than tolerance to global refinement *GCE*. Results can also be put on the time axis (see figure 2) according to year in which their methods were published. Last graph (figure 3) shows statistical quality of each approach corresponding to categorization of the methods.

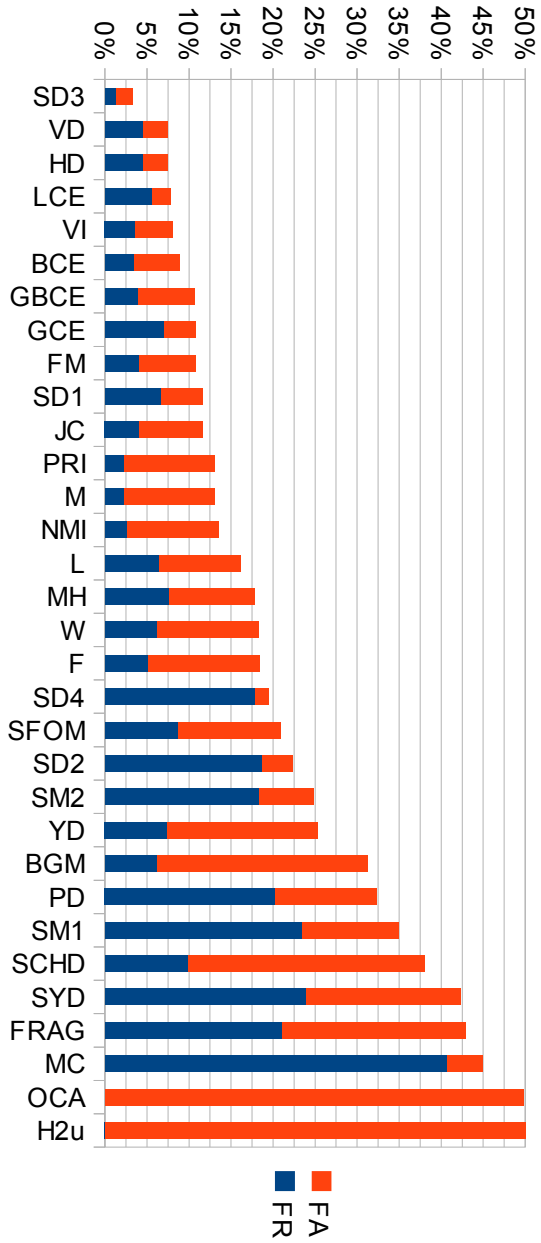


Figure 1: Results of segmentation-segmentation methods for Berkeley data set.

5 CONCLUSION

Segmentation evaluation methods have very various quality. Some of them are focused on a small part of information from the whole segmentation and the quality

 ISBN 978-80-86943-76-3

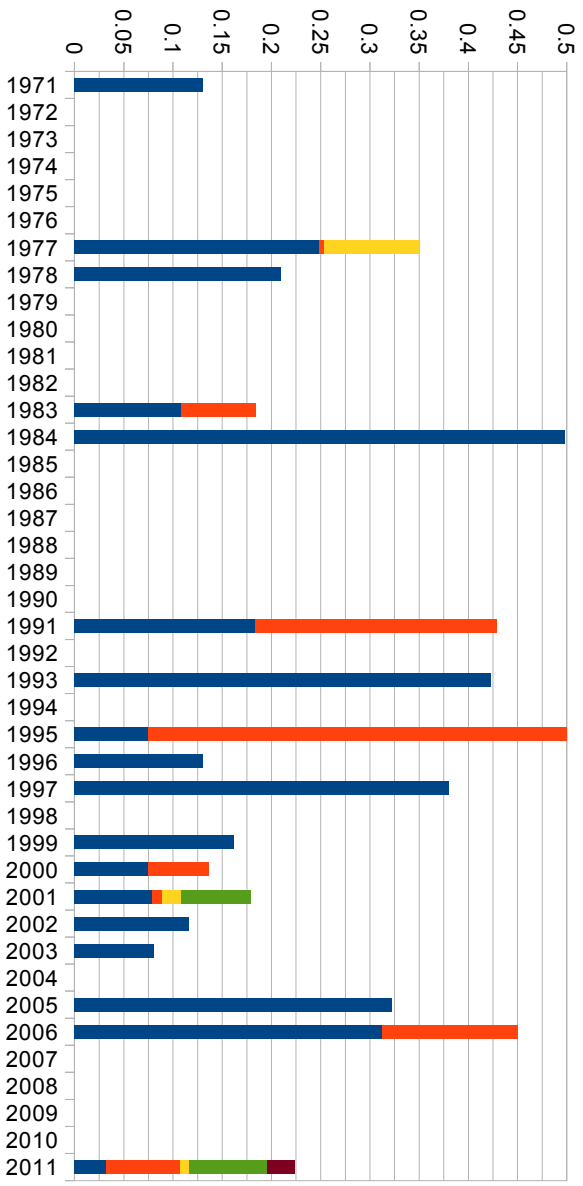


Figure 2: Review of history of methods and evolution of quality. Overall quality is shown only. Methods are stated from left to right. 1971: *PRI*, 1977: SM_2, *YD*, SM_1, 1978: *SFOM*, 1983: *FM*, *W*, 1984: *OCA*, 1991: *F*, *FRAG*, 1993: *SYD*, 1995: *HD*, $H_{2\mu}$, 1996: *M*, 1997: *SCHD*, 1999: *L*, 2000: *VD*, *NMI*, 2001: *LCE*, *BCE*, *GCE*, *MH*, 2002: *JC*, 2003: *VI*, 2005: *PD*, 2006: *BGM*, *MC*, 2011: SD_3, *GBCE*, SD_1, SD_4, SD_2.

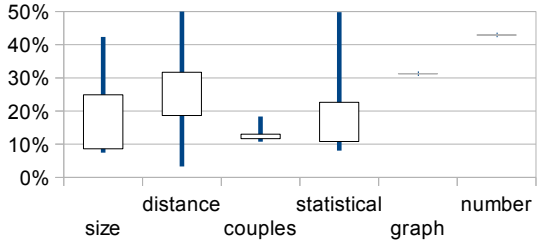


Figure 3: Overview of quality distribution in groups of methods. Minimum, maximum, lower and upper quartiles are presented.

is therefore poor. Even some recently proposed methods do not assure high quality. The best results were provided by method SD_3 which uses grouping of segments and distance measuring. This quality measurement is one of the biggest according to the size of test set as well as the number of methods which can ensure high level of objectivity.

6 REFERENCES

[BHEG02] Asa Ben-Hur, Andre Elisseeff, and Isabelle Guyon. A stability based method for discovering structure in clustered data. *Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing*, pages 6–17, 2002.

[dSCCR05] Jaime dos Santos Cardoso and Luís Corte-Real. Toward a generic evaluation of image segmentation. *Image Processing, IEEE Transactions on*, 14(11):1773–1782, 2005.

[FM83] Edward B. Fowlkes and Colin L. Mallows. A Method for Comparing Two Hierarchical Clusterings. *Journal of the American Statistical Association*, 78(383):553–569, September 1983.

[HD95] Qian Huang and Byron Dom. Quantitative methods of evaluating image segmentation. In *Image Processing, 1995. Proceedings., International Conference on*, volume 3, pages 53–56 vol.3, 1995.

[JMIB06] Xiaoyi Jiang, Cyril Marti, Christophe Irniger, and Horst Bunke. Distance measures for image segmentation evaluation. *EURASIP J. Appl. Signal Process.*, 2006:209–209, January 2006.

[LA99] Bjornar Larsen and Chinatsu Aone. Fast and effective text mining using linear-time document clustering. In *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and*

data mining, KDD '99, pages 16–22, New York, NY, USA, 1999. ACM.

[Mar02] David R. Martin. *An empirical approach to grouping and segmentation*. Phd dissertation, University of California, Berkeley, June-August 2002.

[MC06] Fernando Monteiro and Aurélio Campilho. Performance evaluation of image segmentation. In Aurélio Campilho and Mohamed Kamel, editors, *Image Analysis and Recognition*, volume 4141 of *Lecture Notes in Computer Science*, pages 248–259. Springer Berlin / Heidelberg, 2006.

[Mei03] Marina Meilă. Comparing Clusterings by the Variation of Information. *Learning Theory and Kernel Machines*, pages 173–187, 2003.

[MFTM01] David R. Martin, Charless Fowlkes, Doron Tal, and Jitendra Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. Technical Report UCB/CSD-01-1133, EECS Department, University of California, Berkeley, January 2001.

[MH01] Marina Meilă and David Heckerman. An experimental comparison of model-based clustering methods. *Mach. Learn.*, 42:9–29, January 2001.

[Mir96] Boris G. Mirkin. *Mathematical Classification and Clustering*. Kluwer Academic Publishers, Dordrecht, 1996.

[Pau97] José Paumard. Robust comparison of binary images. *Pattern Recogn. Lett.*, 18:1057–1063, October 1997.

[PB93] Nikhil R. Pal and Dinabandhu Bhandari. Image thresholding: some new techniques. *Signal Process.*, 33:139–158, August 1993.

[Pra78] William K. Pratt. *Digital Image Processing*. John Wiley & Sons, Inc., New York, NY, USA, 1978.

[Ran71] William M. Rand. Objective Criteria for the Evaluation of Clustering Methods. *Journal of the American Statistical Association*, 66(336):846–850, December 1971.

[SG91] Karel C. Strasters and Jan J. Gerbrands. Three-dimensional image segmentation using a split, merge and group approach. *Pattern Recogn. Lett.*, 12:307–325, May 1991.

[SGM00] Alexander Strehl, Joydeep Ghosh, and Raymond Mooney. Impact of Similarity Measures on Web-page Clustering. In *Proceedings of the 17th National Conference on Artificial Intelligence: Workshop of Artificial Intelligence for Web Search (AAAI 2000), 30-31 July 2000, Austin, Texas, USA*, pages 58–64. AAAI, July 2000.

[Sru10] Stepan Srubar. Toward objective segmentation evaluation. In *International Workshops on Computer Graphics, Vision and Mathematics*, 2010.

[Sru11] Stepan Srubar. Quality measuring of segmentation evaluation methods. In *WOFEX*, 2011.

[Sv11] Stepan Srubar and Milan Šurkala. Comparison of mean shift algorithms and evaluation of their stability. In *International Conferences in Central Europe on Computer Graphics, Visualization and Computer Vision (WSCG)*, 2011.

[UPH05] Ranjith Unnikrishnan, Caroline Pantofaru, and Martial Hebert. A measure for objective evaluation of image segmentation algorithms. In *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Workshops - Volume 03*, pages 34+, Washington, DC, USA, 2005. IEEE Computer Society.

[VD00] Stijn Van Dongen. Performance criteria for graph clustering and markov cluster experiments. Technical report, National Research Institute for Mathematics and Computer Science, Amsterdam, Netherlands, 2000.

[Wal83] David L. Wallace. A Method for Comparing Two Hierarchical Clusterings: Comment. *Journal of the American Statistical Association*, 78(383(Sep., 1983)):569–576, 1983.

[YB84] William A. Yasnoff and James W. Bacus. Scene segmentation algorithm development using error measures. In *AOCH 9*, pages 45–58, 1984.

[YMB77] William A. Yasnoff, Jack K. Mui, and James W. Bacus. Error measures for scene segmentation. *Pattern Recognition*, 9(4):217 – 231, 1977.

Java on CUDA architecture

Jan Strnad
Faculty of Information Technology,
Czech Technical University in Prague,
Thákurova 9, 160 00 Prague 6,
The Czech Republic.
strnaj11@fit.cvut.cz

Zdeněk Konfršt
Faculty of Information Technology,
Czech Technical University in Prague,
Thákurova 9, 160 00 Prague 6,
The Czech Republic.
konfrst@fit.cvut.cz

ABSTRACT
This paper presents technologies, programs and Java libraries which allow usage of a CUDA capable GPU device within Java programming language. All these approaches are described with their simplified usage guide. Then, we compare performance of these methods, as well as we compare their friendliness to a programmer, documentation or their maturity. For performance tests, we used matrix multiplication and Gamma correction. We recommend to use JCuda library as currently the best available method, followed closely by JNI.

KEY WORDS
CUDA, GPGPU, GPU, Java, JCuda

INTRODUCTION

Compute Unified Device Architecture (CUDA) is a general-purpose computing on graphics processing units (GPGPU) framework and a hardware architecture by Nvidia. The goal of GPGPU paradigm is to use a graphics processing unit (GPU) as a code execution device. This approach allows us to exploit massively parallel nature of the GPU. The design of GPU completely differs from CPU. GPU has its own memory, which is much faster than the main memory. However, the most important difference is in a GPU core architecture. CUDA GPU cores are generally simpler than CPU cores and they typically run at lower frequencies (around 1 GHz). This simple architecture allows embedding thousands cores per one GPU device.

As mentioned above, CUDA is a technology that allows us to benefit from present GPU architecture. For a detailed description of CUDA, some definitions and brief introduction to the CUDA GPGPU programming please refer to [1].

However, this technology is closely tied with programming language called *C for CUDA*. On the other side, Java is one of the most widely used programming language. In this paper we will present possibilities how to utilize CUDA enabled devices using either standard Java tools or alternatively some third party tools and libraries.

BACKGROUND

In this section we will introduce tools that enable Java to cooperate with CUDA. First tool, which is a part of standard Java distribution, is *Java Native Interface (JNI)*. This tool is "a standard programming interface for writing Java native methods and embedding the Java virtual machine into native applications"[2]. This essentially means that we can implement any Java class or method using C(++) languages. With some tweaks that will follow, C for CUDA may be used as well.

OpenMP/Java [3] and *java-gpu* [4] are both tools that are able to generate a code, that utilizes CUDA device, directly from a Java (byte)code.

JCuda [5], *jCUDA* [6] and *Jacuzzi* [7] are standalone Java libraries. These libraries were designed so that they allow indirect communication of Java code and a GPU. By indirect, we mean that these are only responsible for 'routines' (device initialization, memory transfers between CPU and GPU) but the actual computation is defined elsewhere. In fact, these libraries call a CUDA *kernel* which has to be written in C for CUDA.

AVAILABLE METHODS

In this section we provide relatively detailed description of different approaches that use CUDA device from Java language.

As written above, the **JNI** is probably the most straightforward approach: this technology allows one to implement any Java class method using the C or C++ languages. As a first step we need to define a method to implement in C for CUDA using `native` keyword, for example like this:

```
public native void useCUDA();
```

After that, we are able to generate C header files using `javah` command. The last step is an implementation of functions declared by those header files. These may be implemented using C for CUDA language and thus can utilize a CUDA device. An example of vector addition follows:

```
JNIEXPORT void JNICALL add
```

 ISBN 978-80-86943-76-3

```
  (JNIEnv * env, jobject a,
  jobject b, jobject c)
{
  jclass cls;
  long lptr;//for pointer extraction
  int size;//size of vector
  float * arrayA, *arrayB, *arrayC;
  cs = env->GetObjectClass(a);
  // get required attributes
  get_long_field(env, cs, a, "ptr",
                 &lptr);
  arrayA = (float*)lptr;
  // the same for arrays b and c
  get_int_field(env, cls, a, "size",
                &size);
  vector_add<<<grid, block>>>(
    arrayA, arrayB, arrayC, size);
}
```

When compiling to a shared library, one must make sure that generated code is position independent. This is achieved using following command:

```
$ nvcc -shared -Xcompiler -fPIC -o
    libout.so <src.files>
```

In order to start our program, we must declare where Java should look for our shared native library. This is done using `-Djava.library.path` argument. Detailed description of these steps can be found in [8], page 42.

Second category of available methods uses completely different approach. Unlike the JNI, which is very low-level, this category does not require any knowledge about CUDA and CUDA programming at all. This means that methods from this category take a regular Java code as an input and transform it to a form that uses a CUDA device.

However, these tools do not work fully automatically. The piece of code which should be parallelized must be marked in some way. There are currently two implementations: OpenMP/Java and java-gpu.

Automatic code generation brings some limitations as well. For example, it is not possible to control utilization of fast shared memory. It is also not possible to exploit features such as warp voting etc.

OpenMP/Java is an implementation of Open Multi-Processing (OpenMP) for Java language, which supports not only CPU as a backend, but a CUDA capable GPU as well. This tool is designed as an extension to the standard Java compiler so it can recognize OpenMP commands. These commands are prefixed with symbol `//` (one line comment). That has one positive side effect – the code can be compiled using standard compiler and functionality of that code remains the same. For example, a simple vector addition can be parallelized using single `for` loop:

```
//#omp parallel for
    shared(a,b,result)
for (int i = 0; i < size; i++)
{
  result[i] = a[i] + b[i];
}
```

, where `a`, `b` and `result` are `float` arrays. An OpenMP/Java source can be compile using `jampc` command. It can be run using `jcuda_java` command. Please refer to project's homepage [3] for details.

On the downside, OpenMP/Java contains some bugs. This is mainly due its experimental focus. For instance, we had troubles with compilation and we had to manually alter its source code. Nevertheless, our patch was quite simple – it just converts `int` to `size_t` in one of its functions. Secondly, OpenMP/Java compiler gives us incorrect warnings. Details including installation instruction are available in [8] as well as in [3].

Similar solution called **java-gpu** uses Java annotations to identify a code to be parallelized. However, we were not able to make it work. Specifically, no CUDA code was generated by this tool and CPU was used as the backend instead.

Last tools to mention are Java **libraries** which wrap the Driver API provided by CUDA. This approach is a kind of opposite to the previous one – it requires a very detailed knowledge of CUDA programming. Basically, a source code utilizing a CUDA device is consisted of two parts. First part is written in C for CUDA language and contains only a code which is executed by the GPU device. Second part is written in Java language and it is responsible for a 'glue' code. This for instance include device initialization, memory (de)allocation, memory transfers and naturally CUDA kernel invocations.

Generally, these steps must be done: *a*) Select a device to work with and initialize it. *b*) Create a new CUDA context. *c*) Load a CUDA module – file with the CUDA code to execute(it can be compiled or in PTX format). *d*) Obtain a CUDA function – the CUDA kernel function, which will be used. *e*) Copy data from the main memory to the GPU memory. *f*) Prepare CUDA kernel's arguments. *g*) Invoke the CUDA kernel and wait for a result. *h*) Copy the result back from the GPU memory to the main memory. More details can be found again in [8].

There is currently one up-to-date implementation of the Driver API wrapper: **JCuda**. Other libraries exist, such as **jCUDA** and **Jacuzzi**. However, these are quite outdated and do not support all features provided by newer devices (e.g. surface memory).

EXPERIMENTS

In order to compare performance of different approaches, we designed performance tests. In our

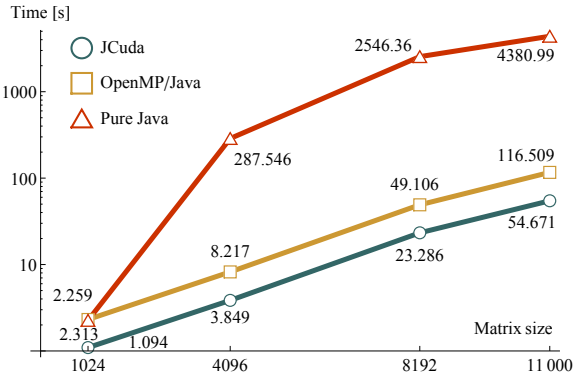


Figure 1: Matrix multiplication performance of different approaches. Note that JNI is covered by JCuda. JNI and JCuda were much faster than OpenMP/Java.

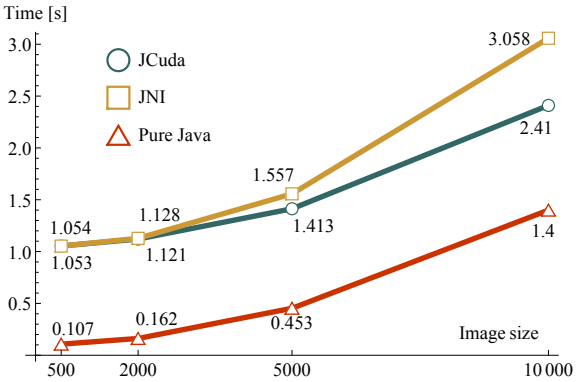


Figure 2: Gamma correction performance of different approaches. OpenMP/Java was not included because it does not support `pow` function.

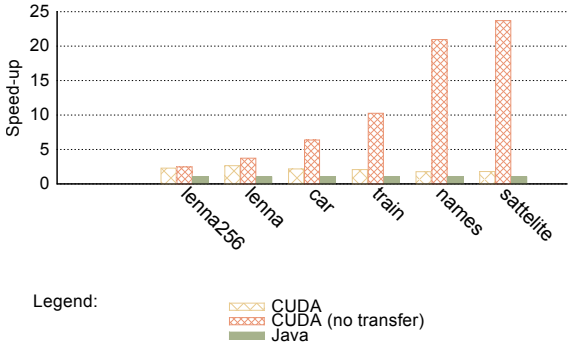


Figure 3: The impact of memory transfers on overall speed-up in Gamma correction algorithm. Memory transfers consumes most of the time.

experiments we used PC with following configuration: Intel Core i5 760 (2.8GHz, 4 cores, 8MB cache), 4GB RAM, Nvidia GeForce GTX 480, Ubuntu 11.04 and CUDA tookit 4.0.

We chose algorithms from linear algebra and image processing for our performance tests. Specifically, we used matrix multiplication and Gamma correction. We used various input data sizes. For linear algebra, we used matrices with width and height from 1024 to 11000, while all matrices were square shaped. Similarly for images which were also square shaped and had sizes from 500 to 10000 pixels. All test were run three times and average value of those was taken into account.

All algorithms which were implemented using JNI and JCuda were optimized in terms of reducing global memory access using shared memory as described in [1]. This was not possible to achieve using OpenMP/Java. The plot also contains pure Java algorithm (single thread) which does not use a GPU at all.

Figure 1 shows performance of different methods for matrix multiplication algorithm. Results of other algorithms from linear algebra can be found in [8].

The plot does not show performance of JNI. The reason is that it would be covered by JCuda otherwise. Those are the two fastest while OpenMP/Java was the slowest. Still, OpenMP/Java was much faster than pure Java.

Results for Gamma correction are in Figure 2. We excluded OpenMP/Java from this test because it lacked of support for `pow` (power) function which was necessary. This plot shows that JCuda was the fastest for Gamma correction. However, all CUDA based algorithm were slower than pure Java. This is not a fault, because programs that use CUDA also contain a device initialization, memory transfers etc. which are included in a result time. Figure 3 shows an impact of memory transfers on overall speed-up in Gamma correction algorithm.

Aside the results of our performance tests, a development time and effort were noted during our experiments. Our experience and some empirical properties (such as operating system support) resulted in an overall subjective method evaluation. To be more specific, every attribute was scored with numbers $1, 2$ and 3 where 1 is the best. All attributes have the same weight. The best approach was determined as the method with minimal sum of scored attributes. See Table 1. Details and table with operating system support can be found in [8].

Attribute	jCUDA	JCuda	OpneMP/Java	JNI
dev. time	2	2	1	3
friendliness	2	1	3	1
docs	2	1	3	1
status	3	1	3	1
Total points	8	5	10	6

Table 1: Comparison of different approaches. JCuda and JNI provide the best support to a programmer while OpenMP/Java the worst.

 ISBN 978-80-86943-76-3

DISCUSSION

All these approaches can be used to improve performance of Java applications. Nevertheless, there were significant performance differences between different methods. To be more specific, tools that transform Java code automatically such as OpenMP/Java could not utilize advanced CUDA specific features e.g. shared memory.

On the other hand, methods like JNI and JCuda were able to utilize all possible features. As mentioned before, performance was not the only one criterion. Other criteria included productivity, stability and documentation. JNI's productivity was very low – this is mostly because even simple operation like copying value of class variable has to be done in about three steps. We have spent at least twice as much programming time when using JNI than with any other method. Since JNI was a part of a standard Java distribution, it does not include any obvious bugs and was documented very well.

OpenMP/Java was a complete opposite of JNI. It had very high productivity (e.g. `for` cycle could be parallelized using one line of code) but it contained some bugs such as incorrect warnings. There was no documentation available. Another evidence of impractical usability is a lack of support for basic mathematic functions such as `pow`. Note that only Linux was supported.

JCuda provides a level of productivity which was generally greater than JNI's but exceptions exist. For example, a device initialization has to be handled manually and also CUDA kernel invocation was more complicated (kernel arguments have to be copied manually). This was caused by a requirement to use the Driver API. JCuda is being developed very actively and it is documented quite well.

CONCLUSION

This paper listed and briefly described solutions for cooperation between CUDA GPGPU technology and Java programming language. These solutions were divided into three categories. First category was JNI. Second category was based on an automatic CUDA code generation from a Java (byte)code. Last solution utilized a Java library which wrapped the standard CUDA Driver API.

JNI and Java libraries were not easy to develop with but they had very good performance. On the other hand, second category was fairly easy to develop with but final performance was not as good as the former. Second category was not production ready yet.

We recommend to use JCuda library. The reason for it is that less development time is required when using JCuda in comparison to JNI. This library is documented quite well, supports various types of operating systems, is production ready and overall performance is identical with JNI. There is currently no other solution which can be both time efficient (in terms of development time) and powerful.

In the matter of future work one can explore another solutions. The first on is Rootbeer. Rootbeer provides additional abstraction so that it can run complex Java objects on a GPU [9].

There are also tools that target more general OpenCL architecture [10] instead of CUDA. There are at least two projects worth trying: JavaCL [11] and Aparapi [12]. Another alternative is ScalaCL that is designed for another JVM language called Scala. This project tries to develop domain specific language that translates to OpenCL code during compilation time [13].

REFERENCES

[1] *CUDA Programming Guide*, Nvidia. [Online]. Available: http://docs.nvidia.com/cuda/cuda-c-programming-guide/index.html

[2] Java Native Interface. Oracle corp. [Online]. Available: http://docs.oracle.com/javase/7/docs/technotes/guides/jni/index.html

[3] M. Philippsen, R. Veldema, M. Klemm, G. Dotzler, and T. Blaß, *OpenMP/-Java*, Friedrich-Alexander University Erlangen-Nuremberg. [Online]. Available: https://www2.informatik.uni-erlangen.de/EN/research/JavaOpenMP/index.html

[4] P. Calvert, *java-gpu*, Computer Laboratory, University of Cambridge. [Online]. Available: http://code.google.com/p/java-gpu/

[5] jcuda.org, *JCuda library*. [Online]. Available: http://jcuda.org/

[6] *jCUDA*, Hoopoe. [Online]. Available: http://www.hoopoe-cloud.com/Solutions/jCUDA/Default.aspx

[7] A. Heusel, *Jacuzzi*. [Online]. Available: http://sourceforge.net/apps/wordpress/jacuzzi/

[8] J. Strnad, "Java on CUDA architecture," Banchelor's thesis, Czech Technical University in Prague, Faculty of Information Technology, 2012. [Online]. Available: https://dip.felk.cvut.cz/browse/pdfcache/strnaj11_2012bach.pdf

[9] T. Kiefer and A. Miftah. Rootbeer GPU compiler. [Online]. Available: http://rbcompiler.com

[10] *OpenCL*, The Khronos Group. [Online]. Available: http://www.khronos.org/opencl/

[11] JavaCL. JavaCL team. [Online]. Available: http://code.google.com/p/javacl

[12] Aparapi. Aparapi team. [Online]. Available: https://code.google.com/p/aparapi

[13] ScalaCL. ScalaCL team. [Online]. Available: https://github.com/ochafik/ScalaCL

GPU-Optimized Bi-Directional Path Tracing

Denis Bogolepov
Lobachevsky State University of Nizhni Novgorod
603950, Nizhni Novgorod, Russia
denisbogol@gmail.com

Danila Ulyanov
Lobachevsky State University of Nizhni Novgorod
603950, Nizhni Novgorod, Russia
danila-ulyanov@ya.ru

Dmitry Sopin
Lobachevsky State University of Nizhni Novgorod
603950, Nizhni Novgorod, Russia
sopindm@gmail.com

Vadim Turlapov
Lobachevsky State University of Nizhni Novgorod
603950, Nizhni Novgorod, Russia
vadim.turlapov@gmail.com

ABSTRACT

In this paper we present a light-weight modification of bi-directional path tracing algorithm that is optimized for massively parallel architectures with limited memory, like GPU. The amount of computations performed by the algorithm is still comparable to unidirectional path tracing. Though modified algorithm preserves some benefits of general bi-directional path tracing and handles indirect illumination and caustics quite efficiently.

Keywords

Realistic Image Synthesis, Interactive Global Illumination, Bi-Directional Path Tracing, GPU, GPGPU.

1. INTRODUCTION

Path tracing is an image synthesis algorithm based on the numerical solution of the rendering equation. This technique allows solving all rendering problems that assume geometric optics, such as soft shadows, indirect lighting, caustics, motion blur, and depth of field. Path tracing provides superior quality visuals compared to rasterization rendering but is also very computationally expensive. Because of the stochastic nature, the image is subject to some variance which is visible as noise. The main contributors to noise are indirect illumination and caustics.

A more general rendering algorithm is *bi-directional path tracing* (BPT) that was independently proposed by Lafortune [Laf93] and Veach [Vea94]. The basic idea is that paths are traced at the same time from a light source and from the camera aperture. All the vertices on the respective paths are then connected using shadow rays and appropriate contributions are added to the measurement of radiance through the corresponding image pixel. BPT handles caustics and indirect illumination effects far more efficiently than ordinary (unidirectional) path tracing (PT).

Despite the fact that the BPT can be implemented on the GPU, it is quite resource intensive. The memory consumption is significantly higher than for ordinary PT (more than 20x for each sample) and depends on the maximum path length [Ant11]. At the same time, effective GPU utilization is achieved for several tens of thousands of concurrent threads that require large amount of onboard memory. However, current GPUs have limited memory resources that should be used sparingly to store 3D geometric models, accelerating structure, texture maps, and other data.

Therefore, we propose a light-weight modification of BPT. The amount of computations performed by the algorithm is still comparable to ordinary PT. Though modified algorithm preserves some benefits of BPT and handles indirect illumination and caustics quite efficiently.

2. THE MEASUREMENT EQUATION

The total radiance M_j measured by the sensor (pixel) j is computed by integrating the incoming radiance L over both the film plane I and all of the surfaces of the scene M:

$$M_j = \int_{I \times M} W_j(\mathbf{x}_1 \to \mathbf{x}_0) L(\mathbf{x}_1 \to \mathbf{x}_0) G(\mathbf{x}_1 \leftrightarrow \mathbf{x}_0) dA_{\mathbf{x}_0} dA_{\mathbf{x}_1}$$

In the equation above, W_j is *response function* that depends on a pixel filter (and/or other factors), and G is *geometry term* defined as

$$G(\mathbf{x}_1 \leftrightarrow \mathbf{x}_0) = V(\mathbf{x}_1 \leftrightarrow \mathbf{x}_0) \frac{\left(N_{\mathbf{x}_0} \cdot \overrightarrow{\mathbf{x}_0\mathbf{x}_1}\right)\left(N_{\mathbf{x}_1} \cdot \overrightarrow{\mathbf{x}_1\mathbf{x}_0}\right)}{\|\mathbf{x}_1 - \mathbf{x}_0\|^2}$$

Here V is the *visibility function* ($V = 1$ if $\mathbf{x}_0$ and $\mathbf{x}_1$ are mutually visible, and $V = 0$ otherwise).

The total amount of outgoing radiance, $L(\mathbf{x}_1 \to \mathbf{x}_0)$, can be computed as the sum of emitted radiance L_e plus reflected radiance L_r:

$$L(\mathbf{x}_1 \to \mathbf{x}_0) = L_e(\mathbf{x}_1 \to \mathbf{x}_0) + L_r(\mathbf{x}_1 \to \mathbf{x}_0) =$$

 ISBN 978-80-86943-76-3

$$L_e + \int_A f_r(\mathbf{x}_2 \to \mathbf{x}_1 \to \mathbf{x}_0) L(\mathbf{x}_2 \to \mathbf{x}_1) G(\mathbf{x}_2 \leftrightarrow \mathbf{x}_1) dA_{\mathbf{x}_2}$$

Here f_r is the *bi-directional scattering distribution function* (BSDF), which describes the reflectance and transmittance properties of a surface.

The last equation is called the *rendering equation* and formulates the law of conservation of light energy in 3D scene. By recursively substituting L on the right side of the rendering equation by the complete right side, we get:

$$L(\mathbf{x}_1 \to \mathbf{x}_0) = L_e(\mathbf{x}_1 \to \mathbf{x}_0) + \sum_{k=2}^{\infty} \int_{M^{k-1}} l(\mathbf{x}_0 \mathbf{x}_1 \mathbf{x}_2 \dots \mathbf{x}_k) dA_{\mathbf{x}_2} \dots dA_{\mathbf{x}_k}$$

Here l: $\bigcup_{i=2}^{\infty} M^{i+1} \to \mathbb{R}$ is the *radiance flow function* defined as

$$l(\mathbf{x}_0 \dots \mathbf{x}_k) = \prod_{i=1}^{k-1} f_r(\mathbf{x}_{i+1} \to \mathbf{x}_i \to \mathbf{x}_{i-1}) G(\mathbf{x}_i \leftrightarrow \mathbf{x}_{i+1}) \cdot L_e(\mathbf{x}_k \to \mathbf{x}_{k-1})$$

This function describes the fraction of radiance from the light source that arrives at the sensor after all of the scattering at vertices between them.

A *light transport path* $\mathbf{X}^k$ is any sequence of surface points $\mathbf{x}_0 \dots \mathbf{x}_k \in I \times M^k$ of which the first point $\mathbf{x}_0$ lies on the film plane I and the remaining lie on the scene surfaces M. The set of all paths of all lengths is called the *path space* and is written as Ω.

The measurement equation can also be written over unified path space Ω. The generalized measurement function, f_j: $\Omega \to \mathbb{R}$, is defined as

$$f_j(\mathbf{x}_0 \dots \mathbf{x}_k) = W_j(\mathbf{x}_1 \to \mathbf{x}_0) \cdot \prod_{i=0}^{k-1} G(\mathbf{x}_i \leftrightarrow \mathbf{x}_{i+1}) \cdot \prod_{i=1}^{k-1} f_r(\mathbf{x}_{i+1} \to \mathbf{x}_i \to \mathbf{x}_{i-1}) \cdot L_e(\mathbf{x}_k \to \mathbf{x}_{k-1})$$

Thus, the measurement of radiance through sensor (pixel) j can be expressed over unified path space as

$$M_j = \int_{\Omega} f_j(\mathbf{X}) d\Omega(\mathbf{X})$$

$$d\Omega(\mathbf{x}_0 \dots \mathbf{x}_k) = dA_{\mathbf{x}_0} \cdot dA_{\mathbf{x}_1} \cdot \dots \cdot dA_{\mathbf{x}_k}$$

This equation formulates the fundamental problem that a global illumination algorithm must solve. To estimate the high-dimensional integral M_j the Monte-Carlo methods are used.

3. GPU-OPTIMIZED BPT

Instant bidirectional path tracing (IBPT) is unbiased rendering algorithm which generates an image in two independent passes (can be executed in any order):

- *Path tracing pass* (PT). Tracing a path starting at the eye (camera lens). The path is extended until it is terminated with a certain probability by Russian roulette. Each vertex $\mathbf{y}_i$ is connected to a random point $\mathbf{z}$ on a light source to form the *explicit view* path $\mathbf{y}_0 \dots \mathbf{y}_i\mathbf{z}$. If the path accidentally hits a light source at point $\mathbf{y}_i$, then the sequence $\mathbf{y}_0 \dots \mathbf{y}_i$ forms *implicit view* path.
- *Light tracing pass* (LT). Tracing a path starting at a selected light source. Each vertex $\mathbf{z}_i$ is directly connected to a random point $\mathbf{y}$ on the camera lens to form the *explicit light* path $\mathbf{z}_0 \dots \mathbf{z}_i\mathbf{y}$. If the path accidentally hits the camera lens at point $\mathbf{z}_i$, then the sequence $\mathbf{z}_0 \dots \mathbf{z}_i$ forms *implicit light* path.

Thus, in IBPT algorithm each path $\mathbf{X}^k = \mathbf{x}_0 \dots \mathbf{x}_k$ can be constructed in four different ways as either an explicit or implicit path on the LT and PT stages. It should be noted that the "implicit light" strategy can be realized only when using the camera with finite aperture (that was implemented in this study).

3.1. Compute Path Contributions

In order to compute the Monte-Carlo contribution of path $\mathbf{X}^k = \mathbf{x}_0 \dots \mathbf{x}_k$, the probability density of sampling this path needs to be expressed with respect to unit surface area. Further the following notation is used to write the densities. Let $p_A(\mathbf{x}_i)$ be a PDF of sampling $\mathbf{x}_i$, measured with respect to unit surface area. The $p_{\omega^\perp}(\mathbf{x}_{i-1} \to \mathbf{x}_i)$ and $p_{\omega^\perp}(\mathbf{x}_i \leftarrow \mathbf{x}_{i+1})$ are the PDFs of sampling $\mathbf{x}_i$ from points $\mathbf{x}_{i-1}$ and $\mathbf{x}_{i+1}$, respectively, measured with respect to projected solid angle. These PDFs relate according to

$$p_A(\mathbf{x}_i) = G(\mathbf{x}_i \leftrightarrow \mathbf{x}_{i+1}) p_{\omega^\perp}(\mathbf{x}_i \leftarrow \mathbf{x}_{i+1}) = G(\mathbf{x}_i \leftrightarrow \mathbf{x}_{i-1}) p_{\omega^\perp}(\mathbf{x}_{i-1} \to \mathbf{x}_i)$$

The probability density of generating a light transport path can be expressed as the product of the sampling densities for the individual path vertices.

PDF of *implicit view* path $\mathbf{X}^k = \mathbf{x}_0 \dots \mathbf{x}_k$ equals

$$p_{V_I}(\mathbf{X}^k) = p_I(\mathbf{x}_0) \cdot \prod_{i=1}^{k} p_A(\mathbf{x}_i) = p_I(\mathbf{x}_0) p_A(\mathbf{x}_1) \cdot \prod_{i=1}^{k-1} G(\mathbf{x}_i \leftrightarrow \mathbf{x}_{i+1}) p_{\omega^\perp}(\mathbf{x}_i \to \mathbf{x}_{i+1})$$

PDF of *explicit view* path $\mathbf{X}^k = \mathbf{x}_0 \dots \mathbf{x}_k$ equals

$$p_{V_E}(\mathbf{X}^k) = p_I(\mathbf{x}_0) \cdot \prod_{i=1}^{k} p_A(\mathbf{x}_i) = p_I(\mathbf{x}_0) p_A(\mathbf{x}_1) \cdot \prod_{i=1}^{k-2} G(\mathbf{x}_i \leftrightarrow \mathbf{x}_{i+1}) p_{\omega^\perp}(\mathbf{x}_i \to \mathbf{x}_{i+1}) \cdot p_A(\mathbf{x}_k)$$

Here $p_I(\mathbf{x}_0)$ and $p_A(\mathbf{x}_1)$ are the probability densities of sampling the vertices $\mathbf{x}_0$ and $\mathbf{x}_1$, respectively. The specific choice of such PDFs is defined by camera model used. For example, finite aperture lens camera is described in [Url01].

PDF of *implicit light* path $\mathbf{X}^k = \mathbf{x}_0 \dots \mathbf{x}_k$ equals

$$p_{L_I}(\mathbf{X}^k) = \prod_{i=0}^{k} p_A(\mathbf{x}_i) = \prod_{i=0}^{k-1} G(\mathbf{x}_i \leftrightarrow \mathbf{x}_{i+1}) p_{\omega^\perp}(\mathbf{x}_i \leftarrow \mathbf{x}_{i+1}) \cdot p_A(\mathbf{x}_k)$$

PDF of *explicit light* path $\mathbf{X}^k = \mathbf{x}_0 \dots \mathbf{x}_k$ equals

$$p_{L_E}(\mathbf{X}^k) = p_I(\mathbf{x}_0) \cdot \prod_{i=1}^{k} p_A(\mathbf{x}_i) = p_I(\mathbf{x}_0) \cdot \prod_{i=1}^{k-1} G(\mathbf{x}_i \leftrightarrow \mathbf{x}_{i+1}) p_{\omega^\perp}(\mathbf{x}_i \leftarrow \mathbf{x}_{i+1}) \cdot p_A(\mathbf{x}_k)$$

For compactness, let $\widehat{W}_j$ denote the *modified sensor response function*:

$$\widehat{W}_j(\mathbf{x}_1 \to \mathbf{x}_0) = \frac{W_j(\mathbf{x}_1 \to \mathbf{x}_0) G(\mathbf{x}_1 \leftrightarrow \mathbf{x}_0)}{p_I(\mathbf{x}_0) p_A(\mathbf{x}_1)}$$

When points on the camera lens and view plane are sampled uniformly, $\widehat{W}_j(\mathbf{x}_1 \to \mathbf{x}_0) = C$ for the simple finite aperture camera model (constant C determines the total sensor sensitivity) [Url01].

After canceling out the common factors, the Monte-Carlo contribution of *implicit view* path $\mathbf{X}^k = \mathbf{x}_0 \dots \mathbf{x}_k$ equals

$$\frac{f_j(\mathbf{X}^k)}{p(\mathbf{X}^k)} = \widehat{W}_j(\mathbf{x}_1 \to \mathbf{x}_0) \prod_{i=1}^{k-1} \frac{f_r(\mathbf{x}_{i+1} \to \mathbf{x}_i \to \mathbf{x}_{i-1})}{p_{\omega^\perp}(\mathbf{x}_i \to \mathbf{x}_{i+1})} \cdot L_e(\mathbf{x}_k \to \mathbf{x}_{k-1})$$

The Monte-Carlo contribution of *explicit view* path $\mathbf{X}^k = \mathbf{x}_0 \dots \mathbf{x}_k$ equals

$$\frac{f_j(\mathbf{X}^k)}{p(\mathbf{X}^k)} = \widehat{W}_j(\mathbf{x}_1 \to \mathbf{x}_0) \prod_{i=1}^{k-2} \frac{f_r(\mathbf{x}_{i+1} \to \mathbf{x}_i \to \mathbf{x}_{i-1})}{p_{\omega^\perp}(\mathbf{x}_i \to \mathbf{x}_{i+1})} \cdot \frac{f_r(\mathbf{x}_k \to \mathbf{x}_{k-1} \to \mathbf{x}_{k-2}) G(\mathbf{x}_k \leftrightarrow \mathbf{x}_{k-1})}{p_A(\mathbf{x}_k)} \cdot L_e(\mathbf{x}_k \to \mathbf{x}_{k-1})$$

The Monte-Carlo contribution of *implicit light* path $\mathbf{X}^k = \mathbf{x}_0 \dots \mathbf{x}_k$ equals

$$\frac{f_j(\mathbf{X}^k)}{p(\mathbf{X}^k)} = W_j(\mathbf{x}_1 \to \mathbf{x}_0) G(\mathbf{x}_1 \leftrightarrow \mathbf{x}_0) \cdot \prod_{i=1}^{k-1} \frac{f_r(\mathbf{x}_{i+1} \to \mathbf{x}_i \to \mathbf{x}_{i-1})}{p_{\omega^\perp}(\mathbf{x}_{i-1} \leftarrow \mathbf{x}_i)} \cdot \frac{L_e(\mathbf{x}_k \to \mathbf{x}_{k-1})}{p_A(\mathbf{x}_k) p_{\omega^\perp}(\mathbf{x}_{k-1} \leftarrow \mathbf{x}_k)}$$

The Monte-Carlo contribution of *explicit light* path $\mathbf{X}^k = \mathbf{x}_0 \dots \mathbf{x}_k$ equals

$$\frac{f_j(\mathbf{X}^k)}{p(\mathbf{X}^k)} = \frac{W_j(\mathbf{x}_1 \to \mathbf{x}_0) G(\mathbf{x}_1 \leftrightarrow \mathbf{x}_0)}{p_I(\mathbf{x}_0)} f_r(\mathbf{x}_2 \to \mathbf{x}_1 \to \mathbf{x}_0) \cdot \prod_{i=2}^{k-1} \frac{f_r(\mathbf{x}_{i+1} \to \mathbf{x}_i \to \mathbf{x}_{i-1})}{p_{\omega^\perp}(\mathbf{x}_{i-1} \leftarrow \mathbf{x}_i)} \cdot \frac{L_e(\mathbf{x}_k \to \mathbf{x}_{k-1})}{p_A(\mathbf{x}_k) p_{\omega^\perp}(\mathbf{x}_{k-1} \leftarrow \mathbf{x}_k)}$$

3.2. Multiple Importance Sampling (MIS)

If there are multiple sampling strategies to generate light transport paths, the samples can be combined in a single unbiased estimator by using MIS.

We propose an efficient procedure for computing the power heuristic weights. Let us consider the inverse weight for *explicit view* path $\mathbf{X}^k = \mathbf{x}_0 \dots \mathbf{x}_k$:

$$\frac{1}{w_{V_E}(\mathbf{X}^k)} = 1 + \frac{p_{V_I}(\mathbf{X}^k)^\beta}{p_{V_E}(\mathbf{X}^k)^\beta} + \frac{p_{L_E}(\mathbf{X}^k)^\beta}{p_{V_E}(\mathbf{X}^k)^\beta} + \frac{p_{L_I}(\mathbf{X}^k)^\beta}{p_{V_E}(\mathbf{X}^k)^\beta} = 1 + \frac{p_{V_I}(\mathbf{X}^k)^\beta}{p_{V_E}(\mathbf{X}^k)^\beta} + \frac{p_{L_E}(\mathbf{X}^k)^\beta}{p_{V_E}(\mathbf{X}^k)^\beta} \left(1 + \frac{p_{L_I}(\mathbf{X}^k)^\beta}{p_{L_E}(\mathbf{X}^k)^\beta}\right)$$

Both explicit and implicit view paths are sampled in the same way except for the last vertex $\mathbf{x}_k$. Thus, the common factors cancel out:

$$\frac{p_{V_I}(\mathbf{X}^k)}{p_{V_E}(\mathbf{X}^k)} = \frac{p_{\omega^\perp}(\mathbf{x}_{k-1} \to \mathbf{x}_k) G(\mathbf{x}_{k-1} \leftrightarrow \mathbf{x}_k)}{p_A(\mathbf{x}_k)}$$

Similar, explicit and implicit light paths differ in how the first vertex $\mathbf{x}_0$ is generated:

$$\frac{p_{L_I}(\mathbf{X}^k)}{p_{L_E}(\mathbf{X}^k)} = \frac{p_{\omega^\perp}(\mathbf{x}_0 \leftarrow \mathbf{x}_1) G(\mathbf{x}_0 \leftrightarrow \mathbf{x}_1)}{p_I(\mathbf{x}_0)}$$

The ratio of the densities of generating explicit light and view paths can be written in the following way:

$$\frac{p_{L_E}(\mathbf{X}^k)}{p_{V_E}(\mathbf{X}^k)} = \frac{1}{p_A(\mathbf{x}_1)} \cdot \frac{1}{p_{\omega^\perp}(\mathbf{x}_1 \to \mathbf{x}_2)} \cdot \frac{p_{\omega^\perp}(\mathbf{x}_1 \leftarrow \mathbf{x}_2)}{p_{\omega^\perp}(\mathbf{x}_2 \to \mathbf{x}_3)} \cdot \dots \cdot \frac{p_{\omega^\perp}(\mathbf{x}_{k-3} \leftarrow \mathbf{x}_{k-2})}{p_{\omega^\perp}(\mathbf{x}_{k-2} \to \mathbf{x}_{k-1})} \cdot p_{\omega^\perp}(\mathbf{x}_{k-2} \leftarrow \mathbf{x}_{k-1}) \cdot p_{\omega^\perp}(\mathbf{x}_{k-1} \leftarrow \mathbf{x}_k) G(\mathbf{x}_k \leftrightarrow \mathbf{x}_{k-1}) = \prod_{i=0}^{k} s_i$$

Obviously, this expression can be computed during the incremental path construction. At the i-th vertex ($i = 0 \dots k$) the following scalar variable needs to be accumulated:

$$s_0 = p_A^{-1}(\mathbf{x}_1)$$
$$s_1 = p_{\omega^\perp}^{-1}(\mathbf{x}_1 \to \mathbf{x}_2)$$
$$s_i = \frac{p_{\omega^\perp}(\mathbf{x}_{i-1} \leftarrow \mathbf{x}_i)}{p_{\omega^\perp}(\mathbf{x}_i \to \mathbf{x}_{i+1})}$$
$$s_{k-1} = p_{\omega^\perp}(\mathbf{x}_{k-2} \leftarrow \mathbf{x}_{k-1})$$
$$s_k = G(\mathbf{x}_k \leftrightarrow \mathbf{x}_{k-1}) p_{\omega^\perp}(\mathbf{x}_{k-1} \leftarrow \mathbf{x}_k)$$

Thus, to compute the inverse MIS weight of explicit view path we only need the accumulated value *s* for the current path vertex. MIS weights for other light transport paths are computed similarly.

3.3. GPU-Specific Features

IBPT can be effectively implemented on massively parallel GPUs due to the following features:

1. Both render passes generate intermediate images independently. Thus, computations can be easily parallelized. In particular, LT and PT passes may be executed on different GPUs (to get the final result two images must be simply summed up).
2. To process paths of any length a constant amount of memory is needed. Therefore, the GPU can be

utilized efficiently by running a large number of light-weight threads. Furthermore, this eliminates restrictions on the unbiased image synthesis.

3. The path connection stage is eliminated (requires information on all vertices of each path).
4. The optimal MIS weights can be computed using only one floating variable per path regardless of its length.

4. RESULTS

The IBPT allows improving the convergence rate (as compared to "ordinary" PT) for regions with indirect illumination problems. Figure 1 shows an example of scene with a significant amount of caustics, which would have taken a long time to render using PT.

Figure 1. Reference image and weighted contributions of LT and PT sampling techniques

The IBPT was compared with the unidirectional MIS PT (combines explicit and implicit view paths). For both rendering algorithms the speed of convergence to the reference image was measured (the difference between images was computed using L2 norm). This comparison is appropriate, since the computational cost of the IBPT is much closer to the ordinary PT than to the general BPT. Strictly speaking, one IBPT sample corresponds to two PT samples. For the test scene, in the region of caustics the IBPT allows to increase the convergence rate by more than an order of magnitude (see Figure 2).

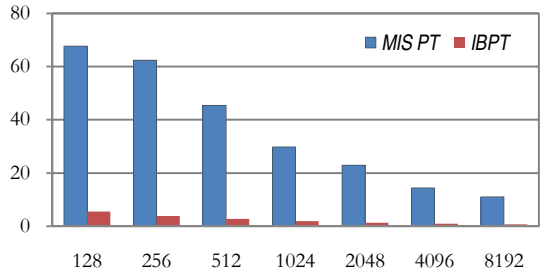


Figure 2. The absolute error depending on the number of samples per pixel (SPP)

Intermediate images generated with a small number of samples per pixel are shown in Figure 3. Bright pixels (fireflies) on the stems of wine glasses are the main factor that limits the convergence speed. These pixels are generated by complicated SDS paths (in Heckbert's notation) that can be processed only by implicit sampling strategies.

Figure 3. PT 128 SPP and IBPT 64 SPP

The general BPT does not solve the problem because the whole set of additional sampling strategies will have zero contribution. For efficient handling of such paths Metropolis Light Transport (MLT) can be used. An alternative approach is the Vertex Merging and Connection algorithm (VCM).

5. CONCLUSION

In this paper we presented light-weight modification of BPT, which is specially optimized for massively parallel architectures with limited memory resources like GPU. We showed that IBPT performs a lot better than simple MIS PT, especially in scenes containing strong indirect illumination and caustics.

6. REFERENCES

[Ant11] D. Antwerpen. *Improving SIMD Efficiency for Parallel Monte Carlo Light Transport on the GPU*. In Proceedings of High Performance Graphics 2011, pp. 41–50, 2011.

[Laf93] E. Lafortune, and Y. Willems. *Bi-Directional Path Tracing*. In Proceedings of Compugraphics '93, pp. 145–153, 1993.

[Vea94] E. Veach, and L. Guibas. *Bidirectional Estimators for Light Transport*. In Fifth Eurographics Workshop on Rendering, pp. 147–162, 1994.

[Url01] D. Antwerpen. *A Survey of Importance Sampling Applications in Unbiased Physically Based Rendering*: http://graphics.tudelft.nl/~dietger/survey.pdf

Calibration of Depth Measurement Model for Kinect-Type 3D Vision Sensors

Branko Karan
Institute of Technical Sciences SASA
Belgrade, Serbia
branko.karan@itn.sanu.ac.rs

ABSTRACT

Accuracy of depth measurement with Microsoft Kinect and similar 3D vision sensors depends on variations in sensor production. Sensor reading may show significant systematic errors that can be compensated in software by using an adequate depth calibration model. This paper presents one such model and a procedure for identification of its parameters. An example calibration is given to illustrate the procedure and the attained improvements.

Keywords

Kinect, 3D sensing, depth camera, RGB-D camera, camera calibration, depth measurement

1. INTRODUCTION

Since its introduction in 2010, Microsoft Kinect sensor gained an enormous popularity in IT community. Prevalent interest has been in its use in computer games and human interface development. Additionally, thanks to its mass production and therefore low market price, there is also a growing interest in using this sensor in other fields, such as robotics (e.g., navigation, fine manipulation) and general 3D sensing.

In emerging fields of applications, accuracy requirements might be generally stronger than with originally envisioned human interfaces. Examples are robotics and similar areas where it is of vital importance to investigate and possibly improve sensor accuracy.

After initial release of Kinect XBOX 360 in 2010, Microsoft has launched an enhanced version named Kinect for Windows in 2012. At the same time, 3D vision sensors with similar design and capabilities have appeared, such as Asus Xtion and PrimeSense Carmine. Thus, it is possible to speak now about Kinect-type sensors that share similar features, both in terms of advantages and drawbacks.

There have been several works dedicated to improvement of accuracy of Kinect-type sensors. Early works, e.g. Burrus [Bur11] and Zhang and Zhang [ZZ11] addressed identification of intrinsic parameters of Kinect cameras. Later, focus was also on refinement of Kinect depth-disparity model, with important works of Khoshelham and Elberink [KE12], Smisek et al. [SJP11], and Herrera C. et al. [HKH12]. These works addressed transformation of Kinect disparity maps into depth maps. However, with new OpenNI [ON13] and Microsoft Kinect SDK [MS13], disparity data are already converted into depth, making the model considered by these authors obsolete.

In this work, an appropriate model for correction of Kinect depth measurements is formulated and a calibration procedure is proposed for identification of model parameters. The procedure is simple in the sense that it relies on commonly adopted camera calibration tools and it does not require specific calibration objects or external measurement devices. The model and the procedure are described in sections 2 and 3. Results of example calibration are given in Section 4, whereas Section 5 summarizes conclusions on the practical aspects of proposed approach.

2. DEPTH MEASUREMENT MODEL

Operation of Kinect depth sensor is grounded on structured light analysis approach. The sensor (Fig. 1) incorporates a laser IR diode for emitting a dotted light pattern and an IR camera for capturing reflected patterns. By using a suitable window size, the sensor compares reflected patterns to reference patterns, obtained for a plane placed at a known distance from the sensor, and uses the position of the best match pattern to infer disparity of reflected bean and further calculate the depth of reflection surface. A supplementary RGB camera is added to provide additional information on color and texture of the surface.

The relationship between depth of reflection surface and the disparity between images of light beans obtained for a reference and measurement (object) surface may be derived in the following manner (the derivation closely follows Khoshelham and Elberink

 ISBN 978-80-86943-76-3

[KE12]). Looking at Fig. 1, where the reference bean is assumed to pass the path $P-R-C$ and the measurement bean passes the path $P-M-C$, from similarity of triangles $MR''C$ and $M'R'C$ we obtain:

$$\frac{d}{D}=\frac{f}{Z} \tag{1}$$

where Z is the distance of the measurement (object) plane from the sensor, $d=\overline{R'M'}$ denotes the disparity between images of reference R' and measurement M' beans, and f is the focal length of IR camera. From similarity of triangles CPR and

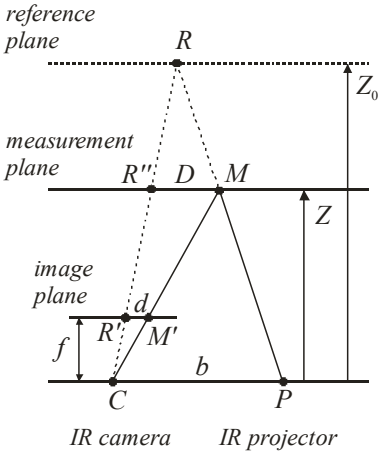


Figure 1. Depth measurement geometry

$R''MR$, another relation is obtained:

$$\frac{D}{b}=\frac{Z_0-Z}{Z_0} \tag{2}$$

in which Z_0 denotes the distance of the reference plane from the sensor. By eliminating D from (1) and (2), the depth measurement model is obtained in the form:

$$\frac{1}{Z}=\frac{1}{Z_0}+\frac{d}{bf} \tag{3}$$

The model (3) is internally used by sensor software to transform detected disparity into depth. However, the value of inferred depth depends on values b, f, Z_0 that are subject to manufacturing variations. Thus, the actual output Z_S from the sensor is really an approximation based on the nominal model:

$$\frac{1}{Z_S}=\frac{1}{Z_{0N}}+\frac{d}{b_N f_N} \tag{4}$$

By combining (3) and (4), the following relationship between Z and Z_S is obtained:

$$\frac{1}{Z}=a_Z\cdot\frac{1}{Z_S}+b_Z \tag{5}$$

where a_Z, b_Z are the values that are characteristics of particular sensor:

$$a_Z=\frac{b_N f_N}{bf};\quad b_Z=\frac{1}{Z_0}-a_Z\frac{1}{Z_{0N}}$$

Ideally, $a_Z=1, b_Z=0$. However, for a specific sensor, depth model parameters a_Z, b_Z may differ from ideal values and this difference may produce significant systematic errors in depth measurement. Therefore, appropriate tuning of depth model parameters may result in improved measurement accuracy.

3. CALIBRATION PROCEDURE

The proposed calibration procedure consists of two steps: the first step comprises standard calibration of sensor's RGB/IR cameras, whereas the calibration of depth model is performed within the second step.

Camera calibration

Camera calibration assumes identification of parameters of functions modeling transformation of coordinates of external objects into coordinates of their images. For a point with homogeneous coordinates $\mathbf{X}_e=\left[X_e,Y_e,Z_e,1\right]^T$ in some external world coordinate frame, transformation of coordinates involves (a) transformation $\mathbf{X}=\mathbf{T}_e\mathbf{X}_e$ into coordinates in camera frame, (b) projection $Z\cdot\mathbf{x}=\left[\mathbf{I}\quad\mathbf{0}\right]\mathbf{X}$ into the point $\mathbf{x}=\left[u,v,1\right]^T$ in normalized image plane, (c) distortion $\mathbf{x}^{(d)}=f^{(d)}(\mathbf{x})$, yielding distorted normalized coordinates $\mathbf{x}^{(d)}$, and finally (d) transformation into pixel coordinates $\mathbf{x}^{(p)}$, using transformation of the form $\mathbf{x}^{(p)}=\mathbf{K}\cdot\mathbf{x}^{(d)}$. Thus, calibration consists in identification of camera matrix $\mathbf{K}$, distortion function $f^{(d)}(\cdot)$ and transformation matrix $\mathbf{T}_e$.

Calibration of Kinect RGB/IR cameras follows a standard procedure of stereo camera calibration whose details is out of scope of this paper. It consists in essence in collecting pairs of images of different views of certain calibration object (e.g., a plane with a checkerboard texture) with known local 3D coordinates of specific feature points (e.g., checkerboard corners). The process proceeds by extracting 2D image coordinates of feature points from individual images and then by performing optimization to find the optimum transformations for all pairs of 3D/2D coordinates of all feature points in all views. The result is obtained in the form of camera matrices $\mathbf{K}_{RGB}$, $\mathbf{K}_{IR}$, distortion functions $f_{RGB}^{(d)}(\cdot)$, $f_{IR}^{(d)}(\cdot)$, and homogeneous transformation matrix ${}^{IR}\mathbf{T}_{RGB}$ for transformation

 ISBN 978-80-86943-76-3

Figure 2. Calibration images

of homogeneous 3D coordinates from RGB to IR camera frame.

To achieve appropriate light conditions for calibration of IR camera, IR emitter has to be disabled during imaging (in this work, the older XBOX 360 was used, so the projector was simply covered by an ordinary stick tape; the newer model Kinect for Windows allows programmable control over the IR emitter).

Good calibration of both RGB/IR cameras is essential part of the procedure, because cameras are used as measurement devices for the second calibration step.

Depth model calibration

During the second step, a series of images is again collected, with each image containing a view to the calibration checkerboard plane placed in different positions and orientations w.r.t. the sensor. However, instead of pairs of RGB/IR images, this time the IR emitter of the sensor is left enabled and a set of pairs of RGB/depth images is collected.

As in the first step, checkerboard corners are used as calibration feature points with known local coordinates in checkerboard frame, which can be expressed as $\mathbf{X}_C(i,j) = \left[(i-1)\cdot w, (j-1)\cdot h, 0, 1\right]^T$, where w, h is the width/height of checkerboard fields. For each view k, the acquired RGB image is transformed into a grayscale image from which 2D pixel coordinates $\mathbf{x}_{RGB}^{(p)}(i,j,k)$ of all checkerboard corners are extracted and paired to known external coordinates $\mathbf{X}_C(i,j)$ to calculate the checkerboard pose ${}^{RGB}\mathbf{T}_C(k)$. Using the known transformation between camera frames, corner coordinates are expressed in IR camera frame as:

$$\mathbf{X}_{IR}(i,j,k) = {}^{IR}\mathbf{T}_{RGB} \cdot {}^{RGB}\mathbf{T}_C \cdot \mathbf{X}_C(i,j)$$

The z-component $Z(i,j,k)$ of such an obtained position is afterward compared to sensor reading. The sensor value is determined by converting $\mathbf{X}_{IR}(i,j,k)$ into pixel coordinates and by searching for the nearest neighbor in sensor depth map m_k:

$$\mathbf{x}_{IR}(i,j,k) = \left\{\begin{bmatrix}\mathbf{I} & \mathbf{0}\end{bmatrix}\mathbf{X}_{IR}(i,j,k)\right\} \Big/ Z(i,j,k)$$
$$\mathbf{x}_{IR}^{(p)}(i,j,k) = \mathbf{K}_{IR} \cdot f_{IR}^{(d)}(\mathbf{x}_{IR}(i,j,k))$$
$$Z_S(i,j,k) = m_k(\text{round}(\mathbf{x}_{IR}^{(p)}(i,j,k)))$$

Finally, the obtained set of pairs $Z(i,j,k)$ and $Z_S(i,j,k)$ is employed to fit the parameters a_Z, b_Z of depth calibration model (5).

4. CALIBRATION EXAMPLE

An experiment has been made to calibrate Kinect using a simple 9×8 checkerboard with 30mm square fields. First, a set of close-up views of the checkerboard placed in different orientations was employed to identify camera parameters. Afterward, for the second phase, a set of 27 RGB/depth image pairs as shown in Figure 2 was prepared. The images were divided in 9 groups, each containing three RGB/depth images of the calibration table placed approximately at the same distance from Kinect but in different orientations. To increase precision, all images were made with the highest resolution of Kinect cameras: 1280x960 for the RGB and 640x480 for the IR camera. For each image acquired by the RGB camera, corner coordinates were then extracted for inner 10×9 corners. In this manner, 9 point clusters, each containing 270 points were generated. The extracted coordinates were used to calculate orientation and position of the table and further to find 3D coordinates of all corners. The values of depth obtained from RGB images are finally compared to depth values obtained from Kinect. The differences are shown in Figure 3, where the points corresponding to particular clusters are displayed in different colors.

It is readily seen that both average values of errors and error deviations obtained from Kinect measurements increase with distance. Moreover, the averages show strong regularity that is in accordance to depth measurement model (5).

Using all collected points, parameters of the depth measurement model were computed by least square fitting as $a_Z = 0.9968$, $b_Z = 4.3651\cdot10^{-6}$. The corresponding correction curve is also shown in Figure 3.

 ISBN 978-80-86943-76-3

Table 1 summarizes statistics on errors obtained before and after correction of depth measurement. It is seen that average values dramatically decreased. Average errors, which were in the range from 1mm for 1m distance (relative error of 0.1%) to 57mm on distance of 3.7m (relative error of 1.5%), practically diminished after calibration, that is, they have been reduced to a level of precision attainable by calibrated

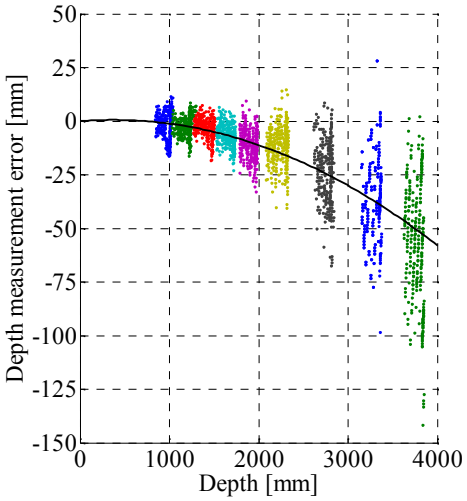


Figure 3. Depth measurement error

RGB camera. On the other side, as expected, confidence intervals (the values in Table 1 are empiric 3σ intervals) underwent only slight changes.

Good matching between experimentally obtained average errors and values predicted by the model offers a possibility to significantly reduce number of calibration measurements. Since the model (5) has only two parameters, it may be expected that similar results could be obtained by using only two clusters of points. Indeed, by selecting clusters 5 and 7, parameters are obtained as $a_Z = 0.9989$, $b_Z = 3.4618 \cdot 10^{-6}$.

Depth [mm]	Depth measurement error [mm]		
	Before calibration	After calibration	Approx. calibration
957	-1.2±18.3	-0.2±18.2	1.0±18.2
1182	-2.4±14.6	-0.1±14.7	1.2±14.7
1411	-3.5±14.2	0.7±13.5	2.0±13.5
1654	-7.0±16.1	-0.3±15.2	0.7±15.3
1907	-11.0±21.1	-1.1±20.5	-0.3±20.6
2228	-11.6±28.1	3.0±28.1	3.3±28.1
2756	-23.2±43.9	1.4±42.7	0.4±42.8
3283	-37.0±57.1	0.0±56.7	-2.7±56.7
3761	-56.9±83.6	-6.2±80.1	-11.0±80.6

Table 1. Improvement in depth measurement

The corresponding statistics calculated for all data points is shown in the last column of Table 1.

5. CONCLUSION

This paper has demonstrated that calibration of Kinect depth sensor can be achieved using the same calibration tools that are commonly used in ordinary camera calibration. Moreover, it has been shown that it is enough to perform calibration imaging at only two distances in the expected range of operation.

It is important to underline that the error reduction demonstrated in this paper is indeed a reduction of differences between reading of Kinect depth sensor and depth values obtained using Kinect RGB camera. Therefore, well calibrated camera is a prerequisite for a quality calibration.

6. ACKNOWLEDGMENTS

This work was supported in part by the Serbian Ministry of Education, Science and Technology Development under grants III-44008 and TR-35003.

All the analysis conducted in this work has been performed in Matlab environment, using Bouguet's Camera Calibration Toolbox [Bou10].

7. REFERENCES

[Bou10] Bouguet, J. Camera Calibration Toolbox for Matlab. http://www.vision.caltech.edu/bouguetj/-calib_doc/ (last modified 09.07.2010)

[Bur11] Burrus, N. Kinect calibration. http://nicolas.-burrus.name/index.php/Research/Kinect-Calibration (last modified 07.08.2011)

[HKH12] Herrera C., D., Kannala, J., and Heikkilä, J. Joint depth and color camera calibration with distortion correction. IEEE Trans. Pattern Analysis and Machine Intelligence, Vol. 34, pp. 2058–2064, 2012.

[KE12] Khoshelham, K. and Elberink, S.O. Accuracy and resolution of Kinect depth data for indoor mapping applications. Sensors, Vol. 12, pp. 1437–1454, 2012.

[MS13] Microsoft Corporation. Kinect for Windows SDK. http://msdn.microsoft.com/en-us/library/-hh855347.aspx (accessed 27.03.2013)

[ON13] OpenNI Consortium. OpenNI, the standard framework for 3D sensing, http://www.openni.-org/ (accessed 27.03.2013)

[SJP11] Smisek, J. Jancosek, M., and Pajdla, T. 3D with Kinect. Proc. IEEE Int. Conf. Computer Vision Workshops, Barcelona, Spain, pp. 1154–1160, 2011.

[ZZ11] Zhang, C. and Zhang, Z. Calibration between depth and color sensors for commodity depth cameras. 2011 IEEE Int. Conf. Multimedia and Expo (ICME), pp. 1-6, 2011.

Volume Ray Casting quality estimation in terms of Peak Signal-to-Noise Ratio

Nikolay Gavrilov Vadim Turlapov

Lobachevsky state university of Nizhny Novgorod
Gagarina 23
Russia 603950, Nizhni Novgorod

{gavrilov86, vadim.turlapov}@gmail.com

ABSTRACT

In spite of a large number of techniques aimed for improvement of Direct Volume Rendering (DVR) quality and performance proposed in the literature, there is a lack of approaches for numerical quality estimation of the images obtained by visualization of medical and scientific volumetric datasets. In this paper we propose a method to estimate sampling artifacts in DVR. Using the proposed estimation method we compare different Ray Casting algorithms to expose optimal ones in quality-performance criteria. We also propose method which combines two techniques for sampling artifacts reduction: Preintegrated DVR and Virtual Samplings method. We show that this combination overcomes both basic methods when using local shading or/and tricubic filtering in RC algorithm.

Keywords

Volume Rendering, Ray Casting, Scientific Visualization.

1. INTRODUCTION

Nowadays there are a lot of GPU-based approaches allowing for interactive rendering [EHK*06]. In addition to different acceleration structures to improve rendering performance by empty space skipping, there are many approaches to improve rendering quality without significant performance [EHMDM08], [KHW*09]. Still there is not any method proposed in the literature to estimate the quality of the output generated by RC algorithm, there are no criteria by which we could compare different quality improvement techniques. Mostly researchers simply put images of competing algorithms side by side, appointing the human visual system to be the judge [MHB*00].

In this paper we propose a noise-based method to estimate sampling artifacts of RC. In addition we propose new RC rendering techniques and compare them with the preintegrated rendering method [EHMDM08] in terms of quality and performance.

2. RAY CASTING ARTIFACTS

Methods involving uniform sampling with post-classification invariably miss thin features along the ray path, omitting the desired surface features thus casing *sampling artifacts* [KHW*09]. The regularity of such artifacts has a wood-like appearance, which can be converted to noise by stochastic jittering of ray starting positions or other shifting strategies [Sch05]. For big datasets (of size greater than 512^3) or transfer functions that cause superficial thin slices the optimization of RC algorithm is needed to keep an acceptable rendering quality and performance.

There is also another important type of DVR artifacts which are introduced by the interpolation method – *filtering artifacts*, caused by trilinear interpolation. Unfortunately they cannot be randomized like sampling artifacts, but they can be significantly reduced by using B-splines interpolation [RtHRS08].

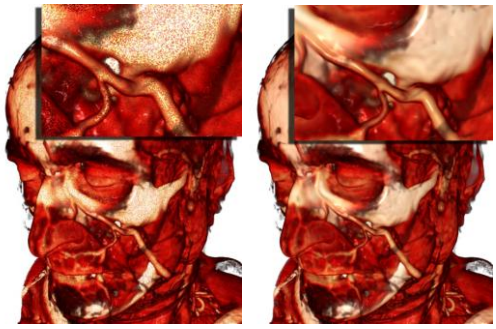


Figure 1. RC quality for sampling rate 1(left) and 8 (right)

Figure 2. Trilinear (left) & tricubic (right) interpolation modes side-by-side comparison.

3. QUALITY MEASUREMENT

As we use stochastic jittering of starting positions of the rays, sampling artifacts can be captured by the

image noise measurement. We can obtain a set of rendered images of the same dataset from the same viewpoint with the same TF and other visualization settings, but we can still change the jittering pattern changing the seed for random values generation.

We can consider the image pixels as a set of independent random values to measure the noise of each single pixel by evaluating the dispersion of its color C (we use YPbPr color space for the calculations below). We consider dispersion of C as an error in pixel (i, j), or as a level of its sampling artifact:

$$D(x,y) = \frac{1}{T}\sum_{i=1}^{T} ||C_i(x,y) - \frac{1}{T}\sum_{j=1}^{T} C_j(x,y)||^2$$

where T is a number of images in the series, $C_i(x,y)$ – color of (x, y) pixel from i^{th} image. It is also possible to use a ground truth image $\tilde{C}$ (image, obtained with a tremendously high sampling rate) instead of the average one:

$$D(x,y) = \frac{1}{T}\sum_{i=1}^{T} ||C_i(x,y) - \tilde{C}(x,y)||^2$$

To measure the noise level we use PSNR (Peak Signal-to-Noise Ratio) logarithmic decibel scale which is mostly used to measure the error introduced by images compression. In the volume rendering domain PSNR is mostly utilized to measure the error of 3d dataset compression [GS04], but not for the rendering quality estimation. We consider PSNR as a quality of the rendering algorithm, it is calculated as follows:

$$PSNR = 10log_{10}\left(\frac{MAX_I^2}{MSE}\right) \qquad MSE = \frac{1}{N}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1} D(i,j)$$

$D(i,j)$ is dispersion (or mean-square error) in (i,j) pixel, MSE – mean-square error of the whole image of size m x n, MAX_I – maximum possible of length pixel in the color space, equal to 1 in our case, N – number of non-background pixels, which is often less than n*m, there are too many background pixels with a null dispersion (see Fig. 4). To avoid the quality overstatement these pixels should be ignored.

We also use PSNR to estimate quality of each single pixel in order to build the quality map of the image which shows us areas of higher and lower rendering quality. The pixels with PSNR > 40dB have no noticeable noise – for these pixels the mean error is less than 1%. Areas where PSNR is less than 30dB contain rather noticeable noise. Figure 3 shows quality maps for different sampling rates.

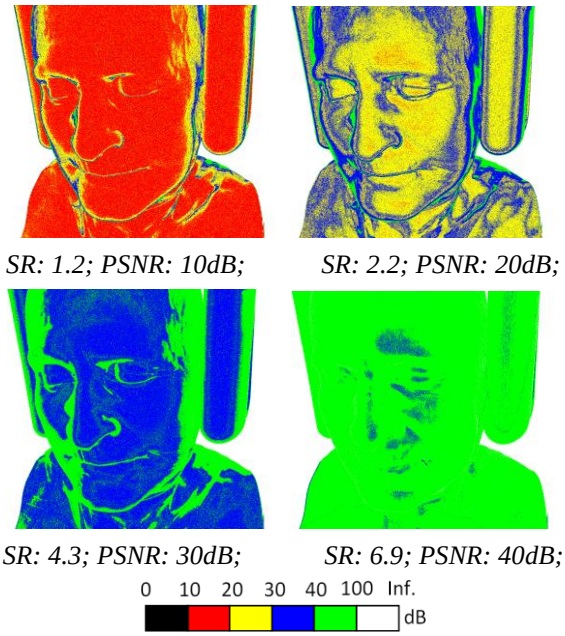


Figure 3. Quality maps and corresponding PSNRs for DVR with different sampling rates (SR).

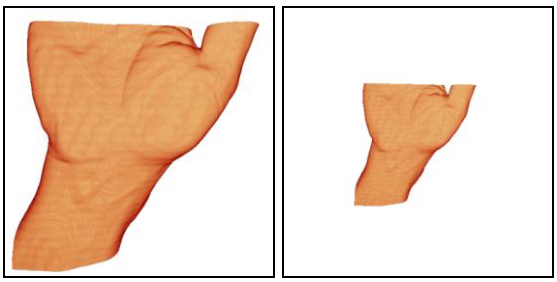


Figure 4. Here we do not ignore background pixels in PSNR calculation.

4. EXPERIMENTS

There are a lot of conditions besides the sampling rate that have influence on RC quality and performance: dataset, viewpoint, transfer function, screen resolution, sizes of bricks we use to decompose our dataset, shading model, filtering method, GPU we use, etc. Fortunately, some of these parameters do not affect rendering quality. Screen resolution affects only the precision of PSNR estimation. The dependence between number of processed pixels and frame rendering time is almost linear, as well as dependence between RC sampling rate and rendering time.

We used 10 different volumetric datasets of sizes 512x512x420 to 512x512x5382 12-bit. We make datasets decomposition into bricks of size 256^3 which appears to be optimal for the GPU we used (GeForce GTX 580 3GB). We used 6 different TFs, some of them impose strict visualization conditions causing presence of thin slices in space.

 ISBN 978-80-86943-76-3

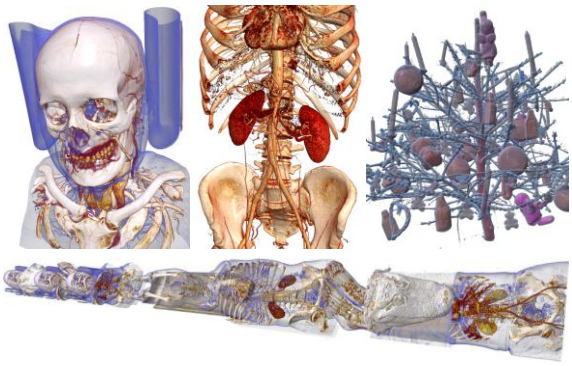


Figure 5. Examples of datasets used in experiments.

Quality-Performance dependence

When comparing efficiency of different quality improvement methods it is necessary to consider not only quality but the rendering performance as well. We change RC sampling rate, thus varying rendering quality and performance. As the result, we obtain quality-performance dependence for each rendering method. Depending on rendering conditions we have obtained a set of dependencies. There are 4 cases: use/not use local shading and trilinear/tricubic filtering. Mostly these two options define the optimal RC algorithm, while other visualization conditions do not have such influence on the efficiency of method. On Figures 6 and 7 we present typical dependencies we obtained in our experiments. Each line corresponds to a rendering method. We vary sampling rate from 1 to 8 samplings per voxel.

Rendering methods we used are: UDVR (Unoptimized DVR), PDVR (Preintegrated DVR [EHMDM08]), CVS (virtual samplings with cubic spline interpolation [LYS*10]), LVS (virtual samplings with linear interpolation), ASM (adaptive step method), PLVS and PCVS (are modifications of LVS and CVS: we make pre-integrated classification instead of post-classification)

In all cases experiments have showed inefficiency of UDVR method. When comparing PSNR values we mostly take in account [30dB, 40dB] range which corresponds to convenient rendering quality. When PSRN>40dB the artifacts are hardly perceptible by a human.

In many cases PDVR height efficiency is caused by its high performance, but when using shading its quality is dramatically limited. ASM is efficient only in cases with shading and trilinear filtering because we perform many samplings causing low performance. On the other hand, we calculate gradient on each ray step thus providing higher shading quality.

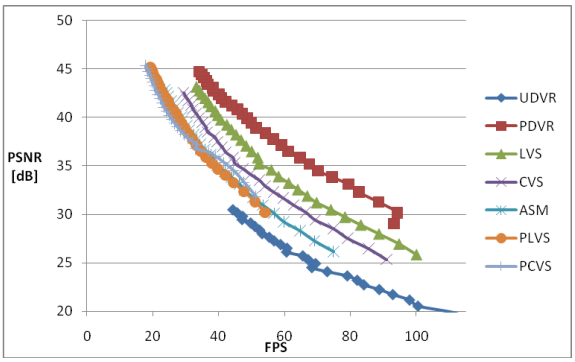


Figure 6. Quality-performance dependencies for different RC algorithms in case of not using tricubic filtering and without local shading.

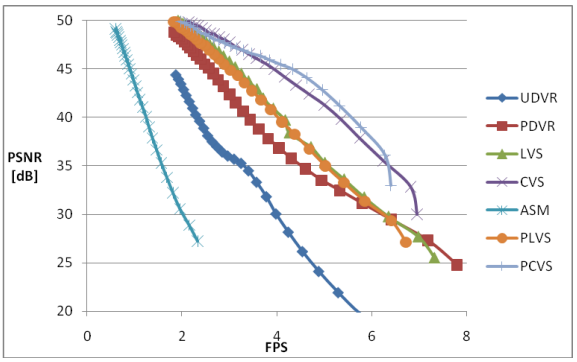
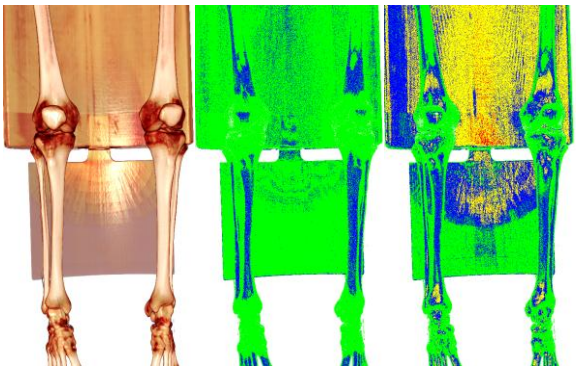


Figure 7. Quality-performance dependencies for different RC algorithms in case of using tricubic filtering and local shading.



DVR Output; PSNR: 36.4dB; PSNR: 22.1dB;

Figure 8. Quality maps for PDVR method without (middle) and with shading (right); sampling rate used here is 2 per voxel.

For LVS and CVS we use interpolated gradient, thus providing acceptable shading quality. But they have lower performance than PDVR has at the same sampling rates and they use post-classification method instead of preintegration, so that often they have lower efficiency.

As for techniques PLVS and PCVS, they are most acceptable in cases of using tricubic filtering or

 ISBN 978-80-86943-76-3

shading. Expensive tricubic filtering compensates the additional computations in these methods to interpolate data values, which are absent in PDVR. On the other hand they provide with better shading. Figure 8 shows low PSNR for the Preintegrated rendering with shading on, while PLVS (or PCVS) methods interpolate gradient to avoid such artifacts.

Optimizing RC algorithms

Before comparing efficiency of RC algorithms we have made their optimization by searching theirs optimal parameters. For instance, in all RC algorithms we have proposed in this paper (all except UDVR and PDVR) there is such parameter as number of ray step division which defines number of internal steps. Varying this parameter for an algorithm we change its quality and performance like we did this by varying the sampling rate. At some point the augmentation of this parameter does not significantly improve quality.

We can also consider a RC method as a set of different RC methods with different number of step divisions in order to build quality-performance dependencies. Figure 9 shows that optimal numbers of divisions in LVS method are 3 and 4. In general we obtain the same result for all other RC methods. However, in cases of TF that causes thin slices in object space we need more step divisions for non-preintegration methods, i.e. LVS, CVS and ASM. In that cases we need up to 10 divisions to avoid severe sampling artifacts. Surely in those cases approaches that use preintegration table work much better.

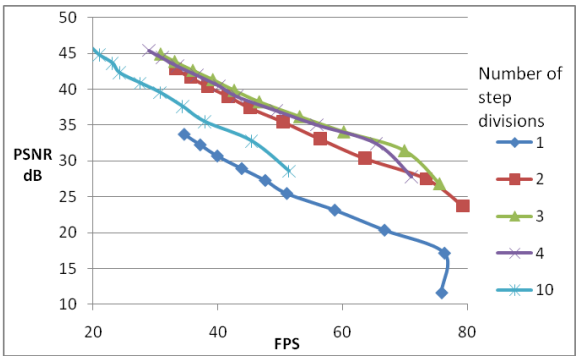


Figure 9. Quality-performance dependencies for LVS algorithm with different number of step divisions.

Shading & Filtering influence on PSNR

Unfortunately filtering artifacts cannot be measured as a noise like sampling artifacts. Still those regions on the image where trilinear filtering artifacts appear have lower PSNR in comparison to those on the image obtained with tricubic filtering. The overall PSNR is almost the same.

The local shading makes the image darker and this causes lower dispersion of intensities of pixels, i.e. higher overall PSNR. Still PDVR method shows better quality when the shading is off.

5. CONCLUSION

A method for Ray Casting quality numerical estimation was proposed. By evaluating noise we calculate PSNR for each single pixel and for the whole image as well. The usage of PSNR allowed us to measure RC noise in decibel scale, and like in images compression domain, the desired quality lies in [30dB, 40dB] range. Comparing PSNR produced by different RC algorithms at fixed fps and varying their parameters we can compute optimal ones for any particular class of visualization cases, e.g. reconstruction filter or shading options.

6. REFERENCES

[EHK*06] Engel K., Hadwiger M., Kniss J., Rezksalama C., Weiskopf D.: Real-time volume graphics. Eurographics Association (2006), 112–114.

[EHMDM08] El Hajjar J. F. et al. Second order pre-integrated volume rendering //Visualization Symposium, 2008. PacificVIS'08. IEEE Pacific. – IEEE, 2008. – C. 9-16.

[GS04] Guthe S., Strasser W.: Advanced techniques for high-quality multi-resolution volume rendering. Computers & Graphics 28, 1 (2004), 51–58.

[KHW*09] Knoll A., Hijazi Y., Westerteiger R., Schott M., Hansen C., Hagen H.: Volume ray casting with peak finding and differential sampling. Visualization and Computer Graphics, IEEE Transactions on 15, 6 (2009), 1571–1578.

[MHB*00] Meissner M., Huang J., Bartz D., Mueller K., Crawfis R.: A practical evaluation of popular volume rendering algorithms. In Proceedings of the 2000 IEEE symposium on Volume visualization (2000), ACM, pp. 81–90.

[LYS*10] Lee, B., Yun, J., Seo, J., Shim, B., Shin, Y. G., & Kim, B. (2010). Fast high-quality volume ray casting with virtual samplings. Visualization and Computer Graphics, IEEE Transactions on, 16(6), 1525-1532.

[RtHRS08] Ruijters D., Ter Haar Romeny B., Suetens P.: Efficient gpu-based texture interpolation using uniform b-splines. Journal of Graphics, GPU, and Game Tools 13, 4 (2008), 61–69.

[Sch05] SCHARSACH H.: Advanced gpu raycasting. Proceedings of CESCG 5 (2005), 67–76.

Efficient simulation of contacts,friction and constraints using a modified spectral projected gradient method

Alessandro Tasora
Università degli Studi di
Parma
Dipartimento di
Ingegneria Industriale
V.le delle Scienze 181/A
Italy (43100), Parma
alessandro.tasora@unipr.it

ABSTRACT

This work introduces a modified version of the Spectral Projected Gradient method that can be used for simulating dynamical systems with complex joints and frictional contacts. The proposed method is able to solve for unknown reactions in systems with large number of colliding shapes and articulated mechanisms. This method couples the ability of solving complementatity constraints, typical of fixed point iterations used in real-time applications, with the superior convergence of Krylov iterations for linear problems, hence making it attractive as a general purpose solver for both linear and non linear problems.

Keywords

Contacts, simulation, complementarity, spectral gradient.

1 INTRODUCTION

Motion of rigid parts with frictional contacts and impacts can be described by Measure Differential Inclusions (MDI), where the non-smooth equations of motion include set-valued forces [Mor99]. The non-smooth approach is expecially appreciated in applications that call for efficient and real-time performances, such as in video games, virtual reality, etc.

Among the methods to perform the time integration of MDI problems, we refer to the approach of [AT10], that expresses the problem as a sequence of Variational Inequalities (VI) to be solved at each time step.

The problem of Variational Inequalities (VI) is stated as the problem of finding $\mathbf{x}$ subject to:

$$\mathbf{x} \in \mathscr{K} \quad : \quad \langle \mathbf{g}(\mathbf{x}), \mathbf{y}-\mathbf{x} \rangle \geq 0 \quad \forall \mathbf{y} \in \mathscr{K} \qquad (1)$$

where $\mathscr{K} \subset \mathbb{R}^n$ is a closed and convex set, and $\mathbf{g}(\mathbf{x}) : \mathscr{K} \to \mathbb{R}^n$ is a continuous function.

Alternatively, Eq.(1) can be formulated using normal cones:

$$\mathbf{x} \in \mathscr{K}, \ \mathbf{g}(\mathbf{x}) \in -\mathsf{N}_{\mathscr{K}}(\mathbf{x}) \qquad (2)$$

We recall the definition of a normal cones $\mathsf{N}_{\mathscr{K}}(\mathbf{x})$ to the set $\mathscr{K}$ at the point $\mathbf{x}$ as:

$$\mathsf{N}_{\mathscr{K}}(\mathbf{x}) = \{\mathbf{y} \in \mathbb{R}^n : \langle \mathbf{y}, \mathbf{x}-\mathbf{z} \rangle \geq 0, \forall \mathbf{z} \in \mathscr{K}\} \qquad (3)$$

A special VI of particular interest in mechanics happens when $\mathbf{g}(\mathbf{x})$ is an affine linear mapping $\mathbf{g}(\mathbf{x}) = A\mathbf{x} + \mathbf{b}$ and when, introducing primal variables $\mathbf{y} \in \mathbb{R}^m$ and dual variables $\mathbf{x}$, we can write:

$$\left\{ \begin{array}{c} \mathbf{0} \\ \mathbf{g}(\mathbf{x}) \end{array} \right\} = \left[\begin{array}{cc} W & J \\ J^T & E \end{array} \right] \cdot \left\{ \begin{array}{c} \mathbf{y} \\ -\mathbf{x} \end{array} \right\} - \left\{ \begin{array}{c} \mathbf{k} \\ \mathbf{c} \end{array} \right\} \qquad (4)$$

Furthermore we introduce the Schur complement

$$A = \left[J^T W^{-1} J - E \right] \qquad (5)$$

$$\mathbf{b} = -\mathbf{c} + J^T W^{-1} \mathbf{k} \qquad (6)$$

for the term $\mathbf{g}(\mathbf{x}) = A\mathbf{x} + \mathbf{b}$ in the VI of Eq.(1).

Following the approach of [AT10], in Eq.(4) the W matrix is the mass matrix, $\mathbf{y}$ represents changes in generalized velocities during Δt, $\mathbf{x}$ are impulses in contacts, $\mathbf{k}$ contains applied impulses, $\mathbf{c}$ is a stabilization term, and E represents compliance, if any. In this class of problems, one solves a VI at each step, where $\mathscr{K}$ in Eq.(4) is

 ISBN 978-80-86943-76-3

the Cartesian product of n_f second-order Lorentz cones $\mathscr{K} = (\bigtimes_{k=1}^{n_f} \mathscr{F}_k)$; each being a Coulomb-Amontons friction cone $\mathscr{F}_k \subset \mathbb{R}^3$. If n_j bilateral constraints are added too (as in mechanical joints), one still has a VI, and the cones associated to the single scalar constraints are just $\mathbb{R}$, thus $\mathscr{K} = (\bigtimes_{k=1}^{n_f} \mathscr{F}_k) \bigtimes \mathbb{R}^{n_j}$. Details can be found in [TA11].

We remark that, as special case, one might have n_j bilateral joints only: this would lead to a simple linear problem because $\mathscr{K} = (\bigtimes_{k=1}^{n_j} \mathbb{R}) = \mathbb{R}^{n_j}$, and $\mathsf{N}_{\mathscr{K}}(\mathbf{x}) = \{\mathbf{0}\}$. For this type of problems, one could use iterative solvers of Krylov type, such as CG or MINRES, that have very fast convergence; however these solvers do not work in case of unilateral and frictional constraints.

On the other side, there are not so many options for the full VI problem. In most cases, projected-Jacobi fixed point iterations are used, as in [TA11, NTM12]. These solvers are very robust, easy to implement, and fit well in real-time scenarios and GPU parallel versions, but their convergence is very poor, especially when one has to simulate articulated shapes and odd mass ratios.

We found that a type of Spectral Projected Gradient (SPG) method can offer the benefits of Krylov iterations and yet retain the broad applicability of the projected-Jacobi fixed point iterations. We will discuss this in the paper.

First, note that the convex set $\mathscr{K}$ is a Cartesian product of low-dimensional convex sets. This means that the projection of a value $\mathbf{x}$ onto the $\mathscr{K}$ set is relatively easy to implement as a sequence of simple projections onto its subsets.

Moreover, with affine $\mathbf{g}(\mathbf{x})$, one can see that Eq.(2) is the first order optimality condition of a Quadratic Program (QP) with convex constraints, where $\mathbf{g}(\mathbf{x}) = \nabla_{\mathbf{x}} f(\mathbf{x})$:

$$\min \quad f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T A\mathbf{x} + \mathbf{x}^T\mathbf{b} \tag{7}$$

$$\text{s.t.} \quad \mathbf{x} \in \mathscr{K} \tag{8}$$

This means that, in order to solve VIs with affine $\mathbf{g}(\mathbf{x})$ and convex $\mathscr{K}$, one can leverage on SPG methods; the attractive feature of the recent SPG methods is that they provide fast enough convergence yet relying only on three computational primitives: product of matrix by a vector, inner products, and projection on convex sets.

Spectral-gradient (SG) methods originated from [BB88] as iterative solvers for unconstrained QPs. A globalization strategy was added in [Ray97] for solving generic non-linear optimization problems. A further advancement was the projected version of solver, that is the SPG presented in [BMR99]. The SPG method solves convex-constrained optimization problems by performing a gradient projection at each step of Barzilai-Borwein iterations. A line search with the Grippo-Lampariello-Lucidi (GLL) strategy can handle the non-monotone nature of the method [GLL86].

We report here our implementation of a preconditioned P-SPG method, similar to the scheme introduced in [BR05], with few modifications. The method requires the following parameters: two safeguards $\alpha_{\min}$ and $\alpha_{\max}$ for the spectral step length (respectively 10^{-9} and 10^9 in our tests), two safeguards for the line search $0 < \sigma_{\min} < \sigma_{\max} < 1$, an integer N_{GLL} (a value about 10 works well in most cases), the Armijo decrease parameter $\gamma \in (0,1)$, and a small value τ_g.

ALGORITHM P-SPG-FB(A, $\mathbf{b}$, $\mathbf{x}_0$, $\mathscr{K}$, $P \mapsto \mathbf{x}$)

$\mathbf{x}_0 := \Pi_{\mathscr{K}}(\mathbf{x}_0)$, $\mathbf{x}_{FB} = \mathbf{x}_0$,
$\hat{\alpha}_0 \in [\alpha_{min}, \alpha_{max}]$
$\mathbf{g}_0 := A\mathbf{x}_0 + \mathbf{b}$, $f(\mathbf{x}_0) = \frac{1}{2}\mathbf{x}_0^T A\mathbf{x}_0 + \mathbf{x}_0^T\mathbf{b}$, $w_0 = 10^{29}$
for $j := 0$ **to** N_{max}
  $\mathbf{p}_j = P^{-1}\mathbf{g}_j$
  $\mathbf{d}_j = \Pi_{\mathscr{K}}(\mathbf{x}_j - \hat{\alpha}_j\mathbf{p}_j) - \mathbf{x}_j$
  if $\langle \mathbf{d}_j, \mathbf{g}_j \rangle \geq 0$
    $\mathbf{d}_j = \Pi_{\mathscr{K}}(\mathbf{x}_j - \hat{\alpha}_j\mathbf{g}_j) - \mathbf{x}_j$
  $\lambda := 1$
  while line search
    $\mathbf{x}_{j+1} := \mathbf{x}_j + \lambda\mathbf{d}_j$
    $\mathbf{g}_{j+1} := A\mathbf{x}_{j+1} + \mathbf{b}$
    $f(\mathbf{x}_{j+1}) = \frac{1}{2}\mathbf{x}_{j+1}^T A\mathbf{x}_{j+1} + \mathbf{x}_{j+1}^T\mathbf{b}$
    if $f(\mathbf{x}_{j+1}) > \max_{i=0,..,\min(j,N_{GLL})} f(\mathbf{x}_{j-i}) + \gamma\lambda\langle \mathbf{d}_j, \mathbf{g}_j \rangle$
      define $\lambda_{\text{new}} \in [\sigma_{\min}\lambda, \sigma_{\max}\lambda]$ and repeat line search
    else
      terminate line search
  $\mathbf{s}_j = \mathbf{x}_{j+1} - \mathbf{x}_j$
  $\mathbf{y}_j = \mathbf{g}_{j+1} - \mathbf{g}_j$
  if j is odd
    $\hat{\alpha}_{j+1} = \frac{\langle \mathbf{s}_j, P\mathbf{s}_j \rangle}{\langle \mathbf{s}_j, \mathbf{y}_j \rangle}$
  else
    $\hat{\alpha}_{j+1} = \frac{\langle \mathbf{s}_j, \mathbf{y}_j \rangle}{\langle \mathbf{y}_j, P^{-1}\mathbf{y}_j \rangle}$
  $\hat{\alpha}_{j+1} = \min(\alpha_{\max}, \max(\alpha_{\min}, \hat{\alpha}_{j+1}))$
  $w_{j+1} = \left|\left| \left[\mathbf{x}_{j+1} - \Pi_{\mathscr{K}}(\mathbf{x}_{j+1} - \tau_g\mathbf{g}_{j+1})\right] / \tau_g \right|\right|_2 = ||\varepsilon||_2$
  if $w_{j+1} \leq \min_{k=0,..,j} w_k$
    $\mathbf{x}_{FB} = \mathbf{x}_{j+1}$
return $\mathbf{x}_{FB}$

This algorithm adds preconditioning, alternate spectral step sizes and a safe fallback strategy to the original SPG method presented in [BMR99].

 ISBN 978-80-86943-76-3

A preconditioned SG is discussed in [LRGH02], and a preconditioned SPG method is presented in [BR05]. By adopting a left-right symmetry-preserving preconditioning with Hermitian $P = LL^T$, exploiting $L^{-T}L^T = I$ and recalling $\mathbf{g} = A\mathbf{x} + \mathbf{b}$, we have $\hat{\mathbf{g}} = \hat{A}\hat{\mathbf{x}} + \hat{\mathbf{b}}$ with

$$\hat{A} = L^{-1}AL^{-T}, \quad \hat{\mathbf{b}} = L^{-1}\mathbf{b}, \quad \hat{\mathbf{x}} = L^T\mathbf{x}, \quad \hat{\mathbf{g}} = L^{-1}\mathbf{g} \tag{9}$$

One can rewrite the original SG method using $\hat{A}$, $\hat{\mathbf{b}}$ and $\hat{\mathbf{x}}$, substituting terms in (9) and alternating two different formulas for the computation of the spectral step size, as suggested in [Fle05]: $\alpha_{BB1} = \langle \mathbf{s},\mathbf{s}\rangle / \langle \mathbf{s},\mathbf{y}\rangle$ and $\alpha_{BB2} = \langle \mathbf{s},\mathbf{y}\rangle / \langle \mathbf{y},\mathbf{y}\rangle$. In the preconditioned case:

$$\hat{\alpha}_{BB1} = \frac{\langle \hat{\mathbf{s}},\hat{\mathbf{s}}\rangle}{\langle \hat{\mathbf{s}},\hat{\mathbf{y}}\rangle} = \frac{\mathbf{s}^T LL^T \mathbf{s}}{\mathbf{s}^T LL^{-1}\mathbf{y}} = \frac{\mathbf{s}^T P\mathbf{s}}{\mathbf{s}^T\mathbf{y}} \tag{10}$$

$$\hat{\alpha}_{BB2} = \frac{\langle \hat{\mathbf{s}},\hat{\mathbf{y}}\rangle}{\langle \hat{\mathbf{y}},\hat{\mathbf{y}}\rangle} = \frac{(L^T\mathbf{s})^T(L^{-1}\mathbf{y})}{(L^{-1}\mathbf{y})^T(L^{-1}\mathbf{y})} = \frac{\mathbf{s}^T\mathbf{y}}{\mathbf{y}^T P^{-1}\mathbf{y}} \tag{11}$$

This has the beneficial effect of partially smoothing the non-monotone descent as shown in Fig.1.

The continuous nonexpansive projection operator $\Pi(\cdot)$ is a mapping that satisfies $\Pi(\mathbf{x})_{\mathscr{K}} = \arg\min_{\mathbf{z}\in\mathscr{K}} ||\mathbf{z}-\mathbf{x}||$. For the preconditioned iteration we use a custom diagonal preconditioner $P = \overline{\mathrm{diag}}(A)$, where the diagonal elements relative to the same sub-set are averaged.

Finally, a fallback strategy is needed because the method, being non monotone, might experience wild oscillations before settling to a stationary point; if one wishes to truncate prematurely the iteration because of real time requirements, as in Fig.2 or in video games, the last computed value $\mathbf{x}_j$ might be really bad. It is wiser to resort to the vector that performed better among those computed; a metric of good performance is $\left|\left|\mathbf{x}_{j+1} - \Pi(\mathbf{x}_{j+1} - \mathbf{g}_{j+1})\right|\right|_2$.

2 RESULTS

We implemented the preconditioned P-SPG-FB method in our C++ library for multibody simulation, in order to solve the dynamic problems with bilateral constraints and frictional contacts. We tested it in various benchmarks and we obtained remarkable convergence properties, as shown in Figs.3,4,5,6.

3 CONCLUSION

In this work we presented the P-SPG-FB method and we discussed its performance in relation to typical scnearios that are found in complex real-time simulations. The fallback strategy cured the problem of the non-monotone nature of the classical SPG method, and preconditioning made it more robust in case of bad conditioning. A remarkable feature is that in case of simple linear problems SPG converges as fast as state-of-the-art CG or MINRES methods. Also, it fits on GPU processor architectures.

4 REFERENCES

[AT10] Anitescu, M., and Tasora, A. *An iterative approach for cone complementarity problems for nonsmooth dynamics*, Computational Optimization and Applications **47(2)** (2010), 207–235.

[BB88] Barzilai, J., and Borwein, J.M., *Two-point step size gradient methods*, IMA Journal of Numerical Analysis **8** (1988), no. 1, 141–148.

[BMR99] Birgin, E.G., Martínez, J.M., and Raydan, M. *Nonmonotone spectral projected gradient methods on convex sets*, SIAM J. on Optimization **10** (1999), 1196–1211.

[BR05] Bello, L., and Raydan, M. *Preconditioned spectral projected gradient method on convex sets*, Journal of Computational Mathematics **23** (2005), 225–232.

[Fle05] Fletcher, R. *On the barzilai-borwein method*, Optimization and Control with Applications (Liqun Qi, Koklay Teo, and Xiaoqi Yang, eds.), Applied Optimization, vol. 96, Springer US, 2005, pp. 235–256.

[GLL86] Grippo, L., Lampariello, F., and Lucidi, S. *A nonmonotone line search technique for newton's method*, SIAM J. Numer. Anal. **23** (1986), 707–716.

[LRGH02] Luengo, F., Raydan, M., Glunt, W., and Hayden, T.L. *Preconditioned spectral gradient method*, Numerical Algorithms **30** (2002), 241–258, 10.1023/A:1020181927999.

[Mor99] Moreau, J.J. *Numerical aspects of the sweeping process*, Computer Methods in Applied Mechanics and Engineering **177** (1999), no. 3-4, 329 – 349.

[NTM12] Negrut, D., Tasora, A., Mazhar, H., Heyn, T., and Hahn, P. *Leveraging parallel computing in multibody dynamics*, Multibody System Dynamics **27** (2012), 95–117, 10.1007/s11044-011-9262-y.

[Ray97] Raydan, M. *The barzilai and borwein gradient method for the large scale unconstrained minimization problem*, SIAM Journal on Optimization **7** (1997), no. 1, 26–33.

[TA11] Tasora, A., and Anitescu, M., *A matrix-free cone complementarity approach for solving large-scale, nonsmooth, rigid body dynamics*, Computer Methods in Applied Mechanics and Engineering **200** (2011), no. 5-8, 439 – 453.

 ISBN 978-80-86943-76-3

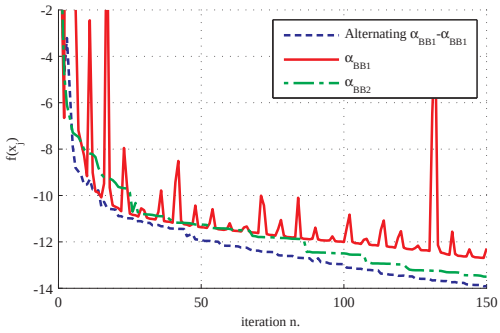


Figure 1: Non-monotone behavior of the SPG method (in benchmark B4).

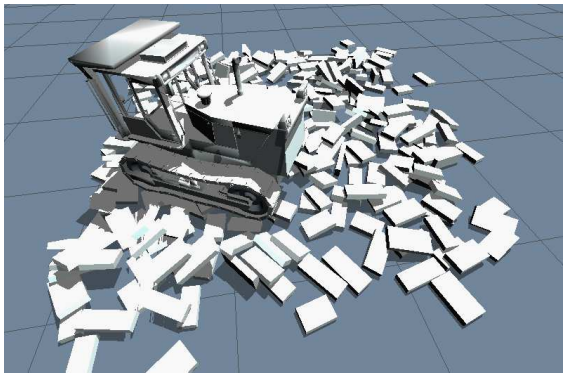


Figure 2: Real-time simulation of a tracked vehicle moving on slabs with rigid frictional contacts.

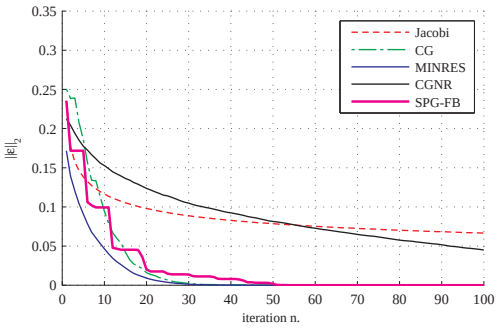


Figure 3: Convergence of benchmark B1: network of 3500 bilateral constraints between 1000 parts.

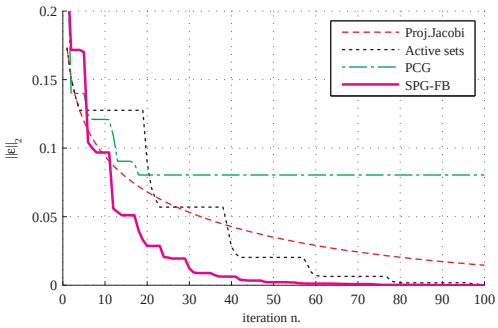


Figure 4: Convergence of benchmark B2: network of 3525 unilateral constraints between 1000 smooth spheres of equal mass.

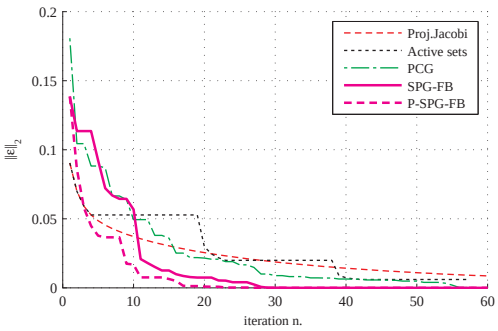


Figure 5: Convergence of benchmark B3: Vertical stack of 20 steel plates, with odd mass ratio.

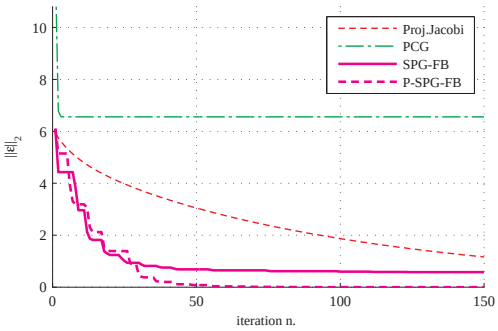


Figure 6: Convergence of benchmark B4: Wrecking ball impacting masonry.

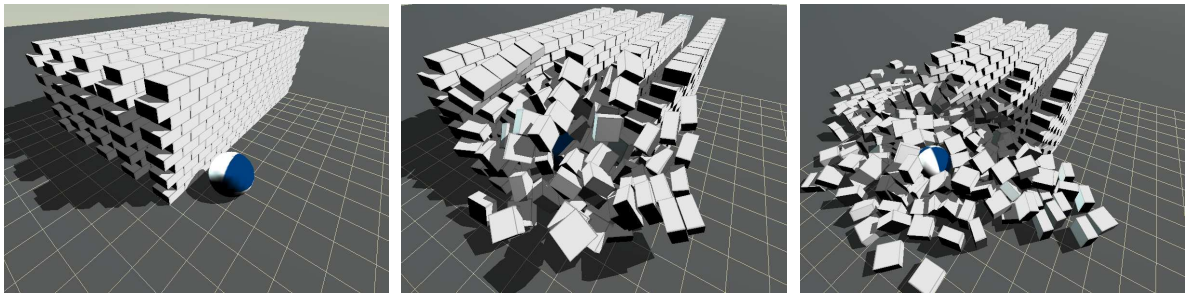


Figure 7: Snapshots from benchmark B4. Simulation of 750 bricks being impacted by a wrecking ball.

 ISBN 978-80-86943-76-3